

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Diseño y elaboración de procedimientos de automatización para procesos en plataformas de planificación de recursos empresariales.

Informe técnico de Residencias Profesionales.

Que para obtener el título de
Ingeniero en Mecatrónica.
Con la especialidad de Automatización.

Presenta
Juan Manuel Reyes Rojas

Coacalco de Berriozábal, Junio/2024

AGRADECIMIENTOS

Agradezco sinceramente a todas las personas e instituciones que han contribuido de manera significativa al desarrollo y culminación exitosa de este proyecto.

En primer lugar, quiero expresar mi profundo agradecimiento al Ingeniero Francisco Javier Sánchez Rico, mi asesor interno del Tecnológico de Estudios Superiores de Coacalco. Su orientación, experiencia y apoyo constante fueron fundamentales para la consecución de este proyecto. Sus valiosas sugerencias y conocimientos enriquecieron cada etapa, guiándome hacia soluciones eficientes y brindándome la confianza necesaria para enfrentar los retos.

Asimismo, agradezco al Ingeniero Eber Guzman Rodríguez, mi asesor externo y jefe directo en Schaeffler Automotive Aftermarket Mexico. Su liderazgo, compromiso y respaldo fueron esenciales para la implementación exitosa de la solución propuesta. Su visión estratégica y apertura a la innovación contribuyeron de manera significativa al logro de los objetivos planteados.

Un reconocimiento especial se dirige al departamento de Materials Management de Schaeffler. La colaboración y el trabajo conjunto con cada miembro de este equipo fueron cruciales para entender a fondo las dinámicas y necesidades específicas del área, permitiendo así diseñar una solución adaptada y eficiente.

Finalmente, agradezco de corazón a mi familia, quienes han sido mi principal apoyo y fuente inagotable de motivación. Su comprensión, paciencia y aliento constante fueron fundamentales durante las largas jornadas de trabajo y estudio. Este logro es también suyo, y celebro este éxito con gratitud por su inquebrantable respaldo.

En conjunto, la contribución de cada persona e instancia mencionada ha sido invaluable, y este proyecto no habría sido posible sin su apoyo y guía. A todos, mi más sincero agradecimiento por ser parte fundamental de este trayecto.

RESUMEN

En el mundo empresarial actual, la eficiencia y la automatización son fundamentales para el éxito empresarial. Este proyecto de investigación titulado "Diseño y desarrollo de métodos de automatización de procesos en plataformas de planificación de recursos empresariales" tiene como objetivo abordar esta necesidad crítica en el contexto de las plataformas de planificación de recursos empresariales (ERP). con especial enfoque en SAP y su implementación en Schaeffler Automotive Aftermarket México.

El principal objetivo de este proyecto es desarrollar procedimientos innovadores que facilitarán la automatización de procesos dentro de las plataformas ERP, optimizando así las operaciones comerciales y administrativas de Schaeffler Automotive Aftermarket México. Se pretende desarrollar soluciones personalizadas que aceleren y simplifiquen tareas complejas, eliminen las intervenciones manuales y limiten los posibles errores humanos. La automatización no sólo mejorará la velocidad y precisión de los procesos, sino que liberará valiosos recursos humanos para actividades estratégicas y creativas, aumentando la productividad y competitividad de la empresa.

Este proyecto se basa en un análisis completo de un proceso existente en la plataforma ERP de la empresa, e identifica áreas específicas que podrían beneficiarse de la automatización. Se utilizan metodologías avanzadas y métodos de integración para diseñar soluciones que sean eficientes, confiables y escalables. Además, se presta especial atención a la seguridad de los datos y la integridad del sistema al implementar soluciones automatizadas.

Una vez finalizado este proyecto, se espera que Schaeffler Automotive Aftermarket México tenga procesos automatizados que no solo mejorarán la eficiencia operativa, sino que también sentarán las bases para una gestión empresarial más ágil, inteligente y adaptable. Estos resultados no sólo beneficiarán a la organización en el corto plazo, sino que también marcarán el estándar para la innovación continua en la optimización de procesos en el campo de las plataformas ERP.

ÍNDICE

GENERALIDADES DEL PROYECTO	7
Introducción.....	8
Descripción de la empresa u organización.	9
Descripción del puesto o área de trabajo.	9
Problema a Resolver	10
Objetivos.....	11
Objetivo general	11
Objetivos específicos	11
Justificación	12
Alcance y limitaciones.....	13
Alcance del Proyecto	13
Limitaciones del Proyecto	14
MARCO TEÓRICO	15
DESARROLLO.....	23
RESULTADOS	61
CONCLUSIONES.....	70
OBJETIVOS EDUCACIONALES	71
FUENTES DE INFORMACIÓN	72
FUENTES DE INFORMACION ILUSTRACIONES	73

ÍNDICE ILUSTRACIONES.

FIGURA 1: SISTEMAS ERP	16
FIGURA 2. VENTANA SAP	17
FIGURA 3.INBOUND	17
FIGURA 4.ÓRDENES DE COMPRA.....	18
FIGURA 5. FACTURAS DE COMPRA	18
FIGURA 6. PANTALLA MENDIX	19
FIGURA 7. VENTANA POWER AUTOMATE.....	20
FIGURA 8. VENTANA VBA.....	21
FIGURA 9. DIAGRAMA DE OPERACIONES	24
FIGURA 10. AUTOMATIZAR OPERACIONES EN SAP	31
FIGURA 11. EJEMPLO DE FACTURA	33
FIGURA 12. INTERFAZ 1.....	33
FIGURA 13. VARIABLES PROGRAMA	35
FIGURA 14: PEDIR NOMBRE DE ARCHIVO	36
FIGURA 15: INPUTBOX.....	36
FIGURA 16: ERROR CON EL NOMBRE DE ARCHIVO	36
FIGURA 17: ASIGNACIÓN DE COLUMNAS.....	37
FIGURA 18: MANEJO DE ERROR TABLA VACÍA.....	37
FIGURA 19: CICLO FOR.....	37
FIGURA 20: COMPARACIÓN ENTRE VALORES.....	38
FIGURA 21: ASIGNACIÓN DE VALORES	38
FIGURA 22: COLOREAR STATUS.....	38
FIGURA 23: OBTENCIÓN DE VALORES	38
FIGURA 24: ASIGNACIÓN DE PROVEEDOR	39
FIGURA 25: INTERFAZ DEL PROGRAMA 2	39
FIGURA 26: COMPARATIVA DE COLUMNAS	40
FIGURA 27: ASIGNACIÓN DE DATOS 2 CASO.....	40
FIGURA 28: CIERRE DE CÓDIGO.....	41
FIGURA 29: VARIABLES PROVEEDOR.....	41
FIGURA 30: SOLICITAR NOMBRE DE DOCUMENTO	42
FIGURA 31: ASIGNACIÓN DE VALORES CON SAP NUMBER.....	42
FIGURA 32: LIMPIAR TABLA.....	43
FIGURA 33: INSTRUCCIONES.....	43
FIGURA 34: INTERFAZ DE USUARIO FINAL	44
FIGURA 35: INTERFAZ DE USUARIO SCOPE.....	44
FIGURA 36: VARIABLES SCOPE	45
FIGURA 37: SOLICITAR NOMBRE SCOPE	45
FIGURA 38: ASIGNACIÓN DE HOJAS SCOPE	46
FIGURA 39: ASIGNACIÓN DE VARIABLES A OBJETOS	46
FIGURA 40: CICLO FOR SCOPE	46
FIGURA 41: NUMERO DE CATALOGO SCOPE	47
FIGURA 42: DIVIDIR CADENA.....	47
FIGURA 43: ASIGNACIÓN DE VALORES SCOPE	47
FIGURA 44: ASIGNACIÓN DE VALORES SCOPE_SUPPLIER.....	48
FIGURA 45: MENSAJE DE ÉXITO SCOPE	48
FIGURA 46: VARIABLES Y NOMBRE DE LIBRO SCOPE	49
FIGURA 47: ESTABLECER OBJETOS SCOPE.....	49

FIGURA 48: ASIGNACIÓN DE VALORES SCOPE CON SAP NUMBERS.....	50
FIGURA 49: LIMPIAR TABLA SCOPE	50
FIGURA 50: USERFORM	51
FIGURA 51: ABRIR EL LIBRO.....	51
FIGURA 52: CÓDIGO USERFORM.....	52
FIGURA 53: SELECCIONAR BASE DE DATOS	52
FIGURA 54: INICIAR EXCEL PA	53
FIGURA 55: CUADRO DE MENSAJE PA	53
FIGURA 56: HOJA DE TRABAJO ACTIVA PA.....	54
FIGURA 57: VARIABLES CONSTANTES PA.....	54
FIGURA 58: OBTENER ÚLTIMA FILA PA	55
FIGURA 59: ALMACENAR COLUMNAS PA	55
FIGURA 60: ABRIR TRANSACCIÓN	55
FIGURA 61: RELLENAR CAMPOS EN LA TRANSACCIÓN.....	56
FIGURA 62: BUCLE PA	57
FIGURA 63: RELLENAR MATERIALES PA.....	57
FIGURA 64: ENVIAR TECLA PA.....	57
FIGURA 65: ALMACENAR ÓRDENES DE COMPRA PA	57
FIGURA 66: CONDICIONAL ÓRDENES DE COMPRA	58
FIGURA 67: MODIFICAR CANTIDAD PA	58
FIGURA 68: CLIC DE SELECCIÓN PA.....	58
FIGURA 69: BOTÓN ADOPT ITEMS PA	58
FIGURA 70: ESCRIBIR CANTIDADES	58
FIGURA 71: CONDICIONALES ÓRDENES DE COMPRA PA	59
FIGURA 72: CONDICIONAL ELSE PA	59
FIGURA 73: FINAL DEL BUCLE.....	60
FIGURA 74: FORMULARIO SELECCIÓN BD	62
FIGURA 75: INTERFAZ DE USUARIO SCOPE.....	62
FIGURA 76: INTERFAZ DE USUARIO MM AVAL	63
FIGURA 77: FACTURA PROVEEDOR	63
FIGURA 78: PLANTILLA CON INFORMACIÓN CARGADA.....	64
FIGURA 79: INTRODUCIR BASE DE DATOS	64
FIGURA 80: GENERACIÓN DE SAP NUMBERS.....	65
FIGURA 81: POWER AUTOMATE - FLUJO.....	65
FIGURA 82: TRANSACCION INBOUND NUMBERS SAP	66
FIGURA 83: PROCESO CARGA DE INFORMACIÓN EN LA TRANSACCIÓN.....	66
FIGURA 84: COMPARATIVA ÓRDENES DE COMPRA	67
FIGURA 85: ADOPT ITEMS	67
FIGURA 86: TRANSACCION FINAL.....	¡ERROR! MARCADOR NO DEFINIDO.
FIGURA 87: PROCESO FINALIZADO EN APLICACIÓN	68
FIGURA 88: CREAR ENTREGA ENTRANTE	68

CAPÍTULO I.
GENERALIDADES DEL PROYECTO

Introducción

En el complejo panorama empresarial actual, la eficiencia operativa se ha vuelto fundamental para la supervivencia y el éxito continuado de las organizaciones. En este contexto, las plataformas de Planificación de Recursos Empresariales (ERP) se han convertido en piedras angulares de la gestión empresarial, simplificando operaciones desde la logística hasta la toma de decisiones estratégicas. Este estudio se adentra en el terreno específico de la automatización de procesos dentro de las plataformas ERP, con un enfoque preciso en SAP, el sistema implementado por Schaeffler Automotive Aftermarket México.

La motivación detrás de esta investigación se nutre de la creciente necesidad de Schaeffler AAM de mejorar su eficiencia operativa. En un mundo donde la rapidez y la precisión son esenciales, la automatización emerge como una estrategia vital para agilizar tareas complejas y minimizar errores humanos. La capacidad para ejecutar procesos de negocio sin intervención manual no solo optimiza la velocidad y la precisión, sino que también libera a los empleados para enfocarse en tareas estratégicas, fomentando así la innovación y el crecimiento.

Para abordar esta problemática, este estudio adopta un enfoque híbrido. En primer lugar, se realizará una revisión exhaustiva de la literatura existente sobre automatización en plataformas ERP, analizando las mejores prácticas y las tendencias actuales. Esta revisión proporcionará el contexto necesario para comprender las complejidades y desafíos del proceso de automatización. En segundo lugar, se llevará a cabo un análisis minucioso de lo proceso específico en SAP de Schaeffler AAM.

Es fundamental reconocer que este estudio se enfrentará a ciertas limitaciones. La automatización estará intrínsecamente ligada a las capacidades tecnológicas de la plataforma SAP y a las restricciones presupuestarias de la organización. Las soluciones propuestas se desarrollarán dentro de estos parámetros, buscando maximizar la eficiencia sin comprometer la integridad del sistema.

Este trabajo se estructura en cuatro capítulos interconectados. El primer capítulo ofrece una revisión detallada del estado actual de la automatización en las plataformas ERP, contextualizando el estudio y delineando las tendencias clave. En el segundo capítulo, se realiza un análisis profundo de los procesos existentes en SAP de Schaeffler Automotive Aftermarket México, destacando áreas críticas para la automatización. El tercer capítulo se enfoca en el diseño e implementación de soluciones automatizadas, detallando la metodología y las técnicas empleadas. Finalmente, el cuarto capítulo presenta las conclusiones derivadas del estudio, junto con recomendaciones para futuras investigaciones y aplicaciones prácticas en el ámbito de la automatización en plataformas ERP. En conjunto, este estudio se propone no solo mejorar la eficiencia operativa de Schaeffler Automotive Aftermarket México, sino también establecer un estándar para la innovación continua en la optimización de procesos dentro de las plataformas ERP en el ámbito empresarial contemporáneo.

Descripción de la empresa u organización.

Schaeffler es una empresa multinacional alemana especializada en tecnología automotriz e industrial. Fundada en 1946, la compañía se ha consolidado como un referente global en el desarrollo y fabricación de componentes clave para la industria automotriz, como rodamientos, sistemas de transmisión, soluciones de motor y tecnología de chasis. Su enfoque se extiende también a la industria aeroespacial y a una amplia gama de aplicaciones industriales.

Con una presencia significativa en todo el mundo, Schaeffler se destaca por su innovación tecnológica y su compromiso con la calidad. Su liderazgo se refleja en la constante búsqueda de soluciones que mejoren la eficiencia, la sostenibilidad y el rendimiento en la industria automotriz e industrial. La empresa opera con una mentalidad de vanguardia, invirtiendo en investigación y desarrollo para adaptarse a los desafíos emergentes del mercado y las demandas de una movilidad más sostenible.

Además, Schaeffler se distingue por su enfoque en la digitalización y la conectividad, aprovechando la tecnología para mejorar sus productos y servicios. Esta orientación hacia la innovación se combina con un compromiso sólido con la responsabilidad social y el desarrollo sostenible, posicionando a Schaeffler como un líder reconocido en la industria, tanto por su excelencia tecnológica como por su compromiso con el progreso y la calidad.

Descripción del puesto o área de trabajo.

Como Trainee en el área de Supply Chain en Schaeffler, el rol implica liderar proyectos de automatización y digitalización en diversos sectores de la empresa, no solo limitados al ámbito de Supply Chain, sino también extendiéndose a otras unidades organizativas clave. La responsabilidad principal radica en identificar oportunidades de mejora, desarrollar estrategias innovadoras y ejecutar proyectos que optimicen la eficiencia operativa y la coordinación entre áreas.

Las tareas incluyen la colaboración estrecha con distintos equipos para analizar los procesos actuales que requieren una intervención, identificar áreas de ineficiencia y proponer soluciones tecnológicas. Se espera implementar sistemas y herramientas avanzadas, como software de gestión de inventario, plataformas de automatización y software de planificación, con el fin de reducir costos, mejorar la visibilidad y coordinación en la cadena de suministro, y optimizar los procesos.

Este puesto también implica el análisis de datos, la evaluación de nuevas tecnologías y la capacitación del personal en el uso eficiente de las herramientas implementadas. La colaboración con diferentes áreas funcionales es esencial para asegurar la integración efectiva de soluciones digitales en toda la empresa, contribuyendo así al desarrollo continuo y al éxito de Schaeffler en un entorno empresarial en constante cambio y evolución hacia la digitalización.

Problema a Resolver

En el contexto operativo de Schaeffler Automotive Aftermarket México, una empresa dedicada a la distribución de repuestos automotrices, la eficiencia se erige como una necesidad ineludible para la supervivencia en un mercado altamente competitivo. A pesar de depender del sistema SAP como su plataforma ERP, se enfrentan a desafíos persistentes. Las operaciones manuales aún predominan en áreas cruciales como inventario, ventas y compras, generando retrasos y errores que afectan directamente la productividad.

En este escenario, la propuesta fundamental de este proyecto se centra en la introducción de procedimientos de automatización específicos en el sistema SAP de la empresa. Esta implementación se vuelve esencial para optimizar procesos dentro de la empresa. La necesidad de automatizar estas áreas críticas radica en la urgencia de mejorar la eficiencia operativa y asegurar respuestas rápidas y precisas a las demandas del mercado.

La falta de acción frente a esta necesidad conlleva consecuencias significativas. La persistencia de procesos manuales en un entorno empresarial cada vez más automatizado podría aumentar los errores humanos, llevando a inexactitudes en los registros financieros y desajustes en los niveles de inventario. La falta de agilidad operativa resultaría en demoras en la toma de decisiones, lo que podría impactar negativamente la capacidad de la empresa para adaptarse rápidamente a las tendencias del mercado y a las necesidades de los clientes. En última instancia, la no implementación de soluciones automatizadas pondría en peligro la competitividad y la viabilidad a largo plazo de Schaeffler Automotive Aftermarket México en un mercado empresarial que está en constante evolución y altamente competitivo.

Ante este panorama, este proyecto no solo se presenta como una necesidad operativa inminente, sino también como una oportunidad estratégica para Schaeffler Automotive Aftermarket México. La automatización no solo mejora la eficiencia, sino que también libera recursos humanos para actividades de mayor valor agregado, fomentando la innovación y el crecimiento continuo en un mundo empresarial que exige adaptabilidad y excelencia operativa.

Objetivos

Objetivo general

El objetivo principal de este proyecto es diseñar e implementar procedimientos de automatización altamente efectivos en las plataformas de planificación de recursos empresariales (ERP) de Schaeffler Automotive Aftermarket México. Estos procedimientos estarán específicamente diseñados para mejorar significativamente la eficiencia y la productividad operativa de la empresa. La meta es optimizar un proceso clave, como lo es la generación de Inbounds a través de facturas de compra ante distintos proveedores, a través de soluciones automatizadas y personalizadas. Al lograrlo, se busca no solo reducir los errores y retrasos asociados con las operaciones manuales, sino también liberar recursos humanos para actividades estratégicas, fomentando así la innovación y el crecimiento continuo de la organización en un entorno empresarial dinámico y competitivo.

Objetivos específicos

Análisis Detallado del Proceso: Realizar un análisis minucioso del proceso de generación de Inbounds dentro de las plataformas ERP (SAP) de Schaeffler Automotive Aftermarket México.

Identificación de Oportunidades de Automatización en Generación de Inbounds: Identificar y documentar oportunidades específicas para automatizar el proceso de generación de Inbounds.

Diseño y Desarrollo de Procedimientos de Automatización para Generación de Inbounds: Desarrollar procedimientos de automatización precisos y detallados para el proceso de generación de Inbounds.

Implementación y Pruebas del Proceso Automatizado: Implementar los procedimientos de automatización y realizar pruebas exhaustivas para validar la funcionalidad y la precisión del proceso automatizado.

Evaluación del Impacto de la Automatización en la Eficiencia y Precisión: Recopilar datos cuantitativos para medir la eficiencia operativa, incluyendo la reducción del tiempo de procesamiento y la disminución de errores.

Establecimiento de un Sistema de Supervisión Continua: Implementar un sistema de supervisión continua para monitorear el proceso automatizado de generación de Inbounds.

Justificación

Esta parte del proyecto surge del profundo interés en el campo de la automatización de procesos en el entorno de las plataformas ERP, específicamente enfocado en la generación automatizada de Inbounds para Schaeffler Automotive Aftermarket México. Este interés se fundamenta en varios motivos que se entrelazan en los ámbitos personal, académico y social.

Desde un punto de vista personal, este proyecto representa una valiosa oportunidad para adquirir conocimientos y habilidades fundamentales en programación y automatización de procesos. La complejidad de diseñar procedimientos automatizados para optimizar operaciones empresariales, especialmente en áreas cruciales de la empresa, proporciona un desafío estimulante y una experiencia educativa enriquecedora. La aplicación de tecnologías avanzadas como la programación y el diseño de aplicaciones, o la propia inteligencia artificial en un contexto empresarial real representa un atractivo desafío intelectual y técnico.

En el ámbito académico, este proyecto se alinea perfectamente con los principios fundamentales de la ingeniería mecatrónica. La automatización de procesos es una piedra angular de esta disciplina y representa una oportunidad invaluable para aplicar teorías y conceptos aprendidos en un entorno empresarial práctico. La experiencia práctica en la implementación de sistemas de automatización realzará la comprensión teórica, fomentando un aprendizaje integral y contextualizado.

A nivel social, este proyecto tiene un impacto significativo en la empresa Schaeffler Automotive Aftermarket México y, por extensión, en la comunidad que la compone. Al optimizar la eficiencia operativa y mejorar la precisión en el manejo de materiales y la cadena de suministro, este proyecto contribuirá directamente a la competitividad de la empresa. Este aumento en la eficiencia no solo beneficia a la empresa en términos de rentabilidad, sino que también puede mejorar la calidad del servicio ofrecido a los proveedores, y de una forma indirecta a los propios clientes, lo que puede tener un efecto positivo en la imagen de la empresa en el mercado.

En resumen, este proyecto representa una oportunidad única para aplicar conocimientos académicos en un contexto empresarial real, mientras se contribuye a mejorar la eficiencia y competitividad de una empresa internacional. La intersección entre el aprendizaje académico, la aplicación práctica y el impacto social lo convierte en un proyecto profundamente motivador y relevante en múltiples niveles.

Alcance y limitaciones

Alcance del Proyecto

El alcance de este proyecto se centra en el diseño, desarrollo e implementación de procedimientos de automatización específicos para el proceso de generación de Inbounds dentro de las plataformas ERP (SAP) de Schaeffler Automotive Aftermarket México. Este alcance incluye un análisis detallado de los procesos existentes, la identificación de áreas de mejora y la implementación de soluciones automatizadas para optimizar la eficiencia y precisión en la comparación de órdenes de compra y facturas de compra. El proyecto abordará el uso de tecnologías como la programación y el desarrollo de aplicaciones, herramientas avanzadas, como podría ser la inteligencia artificial para desarrollar soluciones precisas y confiables. La implementación y evaluación de estos procedimientos automatizados también están dentro del alcance, asegurando su correcta integración con el sistema ERP existente y midiendo su impacto en términos de eficiencia operativa.

Además, es importante destacar que este proyecto tiene un alcance potencialmente expansivo dentro del contexto corporativo de Schaeffler. Aunque inicialmente se implementará en Schaeffler Automotive Aftermarket México, se diseñará de manera que pueda ser adaptado y replicado en otras empresas de Schaeffler en diferentes partes del país, o incluso en otros países. Esta flexibilidad y escalabilidad implican que los procedimientos de automatización desarrollados podrían ser implementados en distintos corporativos de Schaeffler en el futuro, ofreciendo una solución estandarizada y efectiva para optimizar procesos similares en otras ubicaciones dentro del grupo corporativo.

Además, el alcance del proyecto trasciende las fronteras de Schaeffler. La solución desarrollada no está limitada exclusivamente a las aplicaciones internas del corporativo, sino que tiene el potencial de ser compartida y utilizada por otras empresas que enfrentan desafíos similares en la gestión automatizada de procesos logísticos. La naturaleza adaptable y escalable de los procedimientos de automatización permite que este proyecto tenga un impacto potencialmente significativo en el ámbito empresarial más amplio, ofreciendo soluciones innovadoras a una gama diversa de organizaciones que buscan mejorar su eficiencia operativa y precisión en la gestión de datos.

Limitaciones del Proyecto

Este proyecto tiene ciertas limitaciones que definen sus fronteras y alcances:

Alcance Tecnológico: Las soluciones de automatización se desarrollarán dentro de las capacidades tecnológicas existentes en Schaeffler Automotive Aftermarket México. Cualquier requisito de hardware o software adicional tiene que ser aprobado primeramente por los directivos del corporativo.

Alcance de Procesos: El enfoque se limita a la automatización del proceso de generación de Inbounds, específicamente en la adquisición de datos proveniente de facturas y su correcto registro en la plataforma ERP de la empresa. Otros procesos dentro del sistema ERP no estarán por el momento considerados para este proyecto.

Alcance Temporal: El proyecto tiene un límite de tiempo definido para el diseño, desarrollo, implementación y evaluación de los procedimientos de automatización. Esto impone restricciones en la complejidad de las soluciones que se pueden implementar dentro del marco temporal establecido.

Alcance de Recursos: Las limitaciones presupuestarias y de recursos humanos pueden afectar la profundidad y la extensión de las soluciones automatizadas. Cualquier requisito de recursos adicionales que exceda los límites establecidos quedará fuera del alcance del proyecto.

Estas limitaciones definen las fronteras del proyecto y establecen las áreas específicas que no serán cubiertas. Cualquier aspecto fuera de estos límites se considerará fuera del alcance de esta investigación y no será abordado en detalle.

CAPÍTULO II

MARCO TEÓRICO

La automatización de procesos empresariales ha sido un tema crucial en la gestión moderna, transformando la forma en que las organizaciones operan y mejoran su eficiencia. En este contexto, los sistemas ERP, como SAP ERP 6.0, han surgido como pilares fundamentales para la integración y coordinación de diversas funciones empresariales.

"Los sistemas ERP son aplicaciones de software que integran y gestionan la mayoría o todas las áreas funcionales de una organización. Proporcionan un conjunto de herramientas para ayudar a las empresas a manejar y supervisar sus actividades diarias, desde la contabilidad y los recursos humanos hasta la gestión de inventarios y la cadena de suministro. Estos sistemas permiten una visión holística de las operaciones empresariales, lo que facilita una toma de decisiones más informada y una mayor eficiencia en toda la organización" (Laudon & Laudon, 2016).



Figura 1: Sistemas ERP

"SAP ERP 6.0 es un sistema de planificación de recursos empresariales desarrollado por SAP AG. Esta plataforma integra diversas funciones empresariales, incluyendo contabilidad, gestión financiera, recursos humanos y gestión de inventarios, en un sistema centralizado. SAP ERP 6.0 proporciona a las organizaciones una visión en tiempo real de sus operaciones y recursos, permitiendo una gestión eficiente y efectiva. Además, ofrece herramientas de análisis avanzado y generación de informes que facilitan la toma de decisiones informada y estratégica para las empresas" (SAP AG, 2019).



Figura 2. Ventana SAP

La implementación de SAP ERP 6.0 permite a las empresas consolidar sus operaciones, desde la gestión de inventario hasta la planificación de recursos humanos, en un único sistema centralizado. Esta integración no solo mejora la eficiencia, sino que también proporciona una base sólida para la automatización de procesos empresariales complejos y cruciales, como la gestión de Inbounds, facturas de compra y órdenes de compra.

"Los Inbounds son los bienes, productos o materiales que una organización recibe de sus proveedores. Estos artículos son esenciales para el funcionamiento continuo de la empresa y pueden incluir materias primas, productos terminados o componentes necesarios para la fabricación. La gestión eficiente de los Inbounds es crucial para mantener una cadena de suministro fluida y garantizar que la empresa tenga suficientes existencias para satisfacer la demanda del mercado" (Bowersox, Closs, & Cooper, 2007).



Figura 3. Inbound

En el ámbito logístico, los Inbounds representan una parte vital de la cadena de suministro, con su gestión eficiente siendo esencial para garantizar una operación fluida y una satisfacción del cliente óptima.

"Las órdenes de compra son documentos legales emitidos por una empresa para formalizar la adquisición de bienes o servicios de un proveedor externo. Las órdenes de compra son esenciales para establecer un acuerdo contractual claro entre la empresa compradora y el proveedor, proporcionando detalles precisos sobre la transacción" (Monczka, Handfield, Giunipero, & Patterson, 2015).



Figura 4. Órdenes de Compra

"Las facturas de compra son documentos comerciales que detallan los bienes o servicios proporcionados por un vendedor a un comprador. Estas facturas contienen información crucial, como la descripción detallada de los artículos vendidos, las cantidades entregadas, los precios unitarios acordados, cualquier descuento aplicado y los términos de pago. " (Van der Meer, 2012).



Figura 5. Facturas de Compra

La automatización de este proceso no se trata solo de optimizar la recepción y el almacenamiento de productos, sino también de crear una red de comunicación ágil con los proveedores.

Esta visibilidad mejora la capacidad de la empresa para tomar decisiones basadas en datos, mejorando la agilidad operativa. La conciliación entre las facturas de compra y las órdenes de compra es una tarea delicada pero fundamental en cualquier empresa. La automatización de este proceso no solo mejora la precisión y reduce los errores, sino que también libera el tiempo del personal para tareas más estratégicas.

Por otro lado, las herramientas como Mendix, Power Automate y Visual Basic for Applications representan un salto adelante en la automatización empresarial.

"Mendix es una plataforma de desarrollo de aplicaciones de bajo código que permite a las organizaciones crear aplicaciones empresariales personalizadas de manera rápida y sencilla. Al proporcionar una interfaz visual intuitiva y herramientas de arrastrar y soltar, Mendix permite a los usuarios crear aplicaciones sin necesidad de codificación extensiva. La plataforma también ofrece funcionalidades avanzadas para la integración de datos, la automatización de procesos y la colaboración en equipo, lo que la convierte en una herramienta poderosa para la transformación digital y la innovación empresarial" (Fico, 2020).

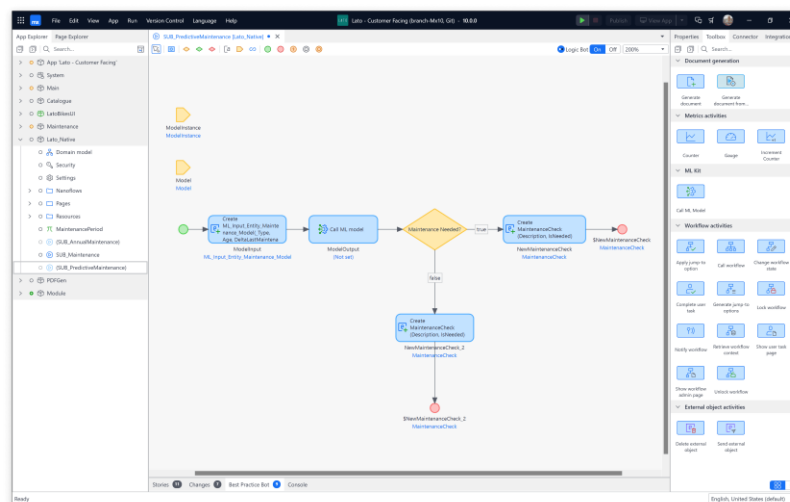


Figura 6. Pantalla Mendix

Mendix, siendo una plataforma sin código, permite a los usuarios crear aplicaciones personalizadas de manera rápida y sin requerir habilidades de codificación avanzadas. Esta capacidad es fundamental para adaptar soluciones a las necesidades específicas de Schaeffler, permitiendo una automatización ágil y altamente personalizada de los procesos de negocio.

"Power Automate es una plataforma de automatización de flujos de trabajo desarrollada por Microsoft. Permite a los usuarios crear flujos de trabajo automatizados que conectan diversas aplicaciones y servicios, tanto en la nube como locales, para facilitar la automatización de tareas y procesos empresariales. La plataforma ofrece una interfaz intuitiva y basada en el navegador que permite a los usuarios diseñar flujos de trabajo sin necesidad de habilidades de codificación avanzadas. Power Automate permite la integración sin problemas entre aplicaciones populares como Microsoft 365, SharePoint, Dynamics 365, y muchas otras, lo que facilita la automatización de procesos complejos y la mejora de la colaboración y la eficiencia en toda la organización" (Floyd, 2021).

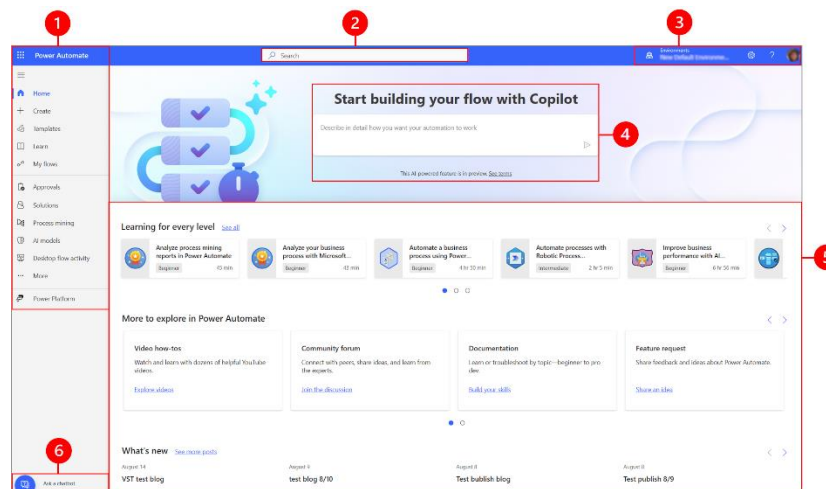


Figura 7. Ventana Power Automate

Power Automate, facilita la conexión entre diversas aplicaciones y servicios, permitiendo flujos de trabajo complejos y altamente automatizados.

Finalmente, la integración de Visual Basic for Applications (VBA) con SAP ofrece una oportunidad única para la personalización y la automatización avanzada. VBA no solo permite la creación de scripts personalizados para SAP, sino que también puede interactuar con otras herramientas y sistemas utilizados por Schaeffler.

"Visual Basic for Applications (VBA) es un lenguaje de programación desarrollado por Microsoft. Diseñado como una extensión del popular lenguaje de programación Visual Basic, VBA se utiliza para automatizar tareas y procesos dentro de diversas aplicaciones de Microsoft, como Excel, Word, Outlook y Access. Los usuarios pueden escribir scripts en VBA para crear macros, que son secuencias de instrucciones que automatizan acciones específicas en estas aplicaciones. Estas macros pueden ser utilizadas para realizar tareas repetitivas, manipular datos, generar informes y mucho más, lo que mejora significativamente la eficiencia y la productividad en el entorno empresarial" (Walkenbach, 2015).

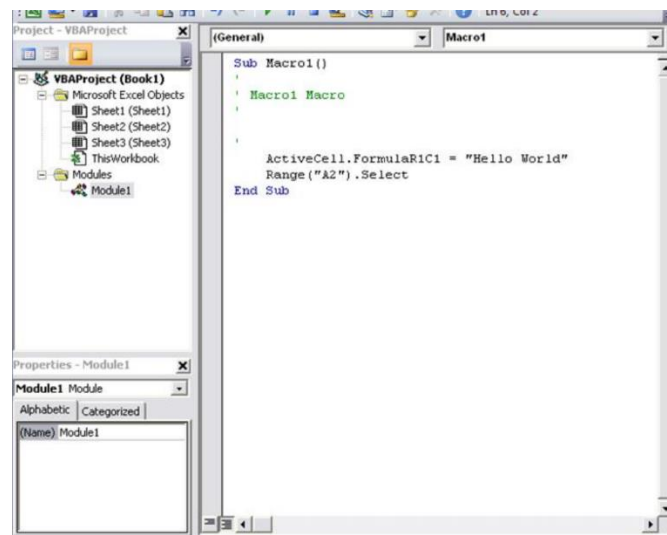


Figura 8. Ventana VBA

La capacidad de VBA para automatizar tareas, combinada con su habilidad para integrarse con otras soluciones empresariales, crea flujos de trabajo completamente automatizados y sin problemas.

Esta combinación de herramientas proporciona a Schaeffler una capacidad inigualable para automatizar y optimizar sus procesos, mejorando tanto la eficiencia como la precisión en sus operaciones diarias.

Esta integración profunda no solo mejora la eficiencia operativa, sino que también proporciona una experiencia del usuario coherente y fluida a través de todas las plataformas y sistemas utilizados por la empresa.

La necesidad de automatizar los procesos de gestión de Inbounds se hace evidente. Al analizar los sistemas ERP como SAP ERP 6.0, se revela su papel esencial como columna vertebral en la integración y gestión eficiente de datos empresariales. Automatizar la gestión de Inbounds además de disminuir notablemente el tiempo invertido en esta actividad, permite decisiones basadas en datos precisos y actualizados de una manera mucho más sencilla. De manera similar, no solo implica reducir errores, sino también liberar recursos para tareas estratégicas, al tiempo que contribuye a una gestión financiera más precisa y planificación estratégica. La integración de herramientas como Mendix, Power Automate o VBA con los sistemas ERP presenta oportunidades para optimizar aún más estos procesos, proporcionando flujos de trabajo fluidos y eficientes que mejoran tanto la operatividad como la experiencia del usuario en Schaeffler Automotive Aftermarket México. Estas soluciones no solo optimizan los flujos de trabajo, sino que también sientan las bases para una toma de decisiones más informada y una gestión empresarial más ágil y adaptable en el dinámico mercado actual.

CAPÍTULO III

DESARROLLO

Primeramente, se comenzó a realizar un análisis de todo el proceso de generación de los Inbounds dentro de Schaeffler Automotive Aftermarket México. Para ello se realizaron ciertas capacitaciones con el personal especializado de Materials Management, que es el área dentro de la empresa que se encarga de realizar estas operaciones.

Para simplificar el entendimiento se decidió realizar un diagrama que permitiera mostrar paso a paso cada parte del proceso.

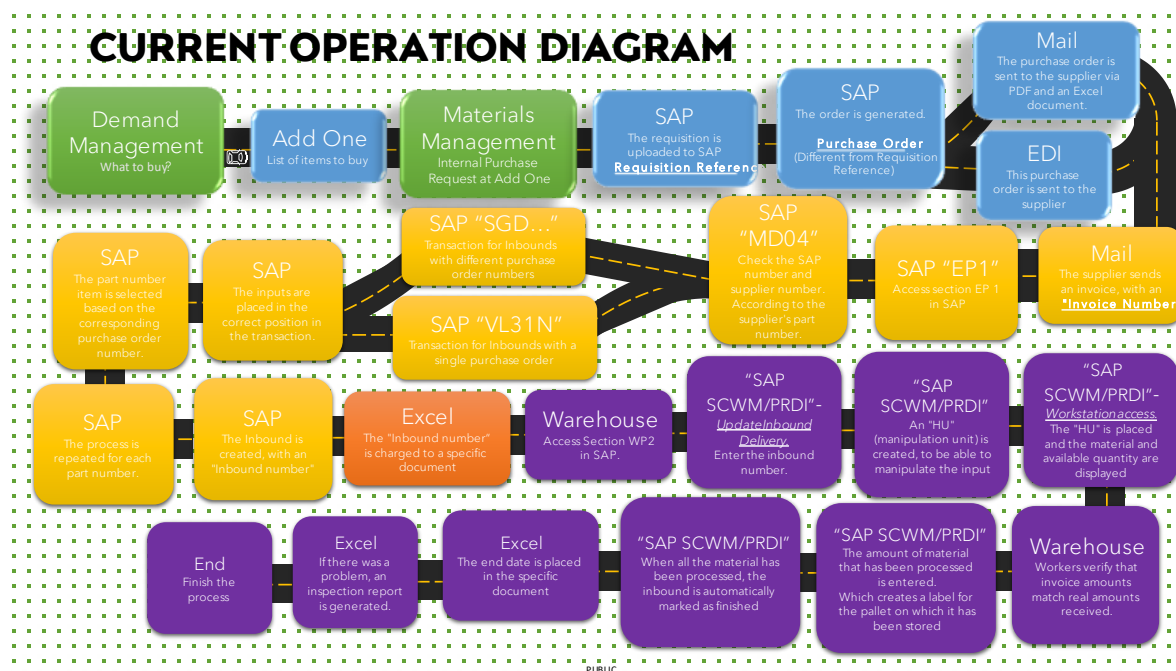


Figura 9. Diagrama de Operaciones

En la Figura 9, se puede observar todo el proceso, comienza con otro departamento de la empresa, llamado “Demand Management”, que mediante un sistema de gestión empresarial llamado Add-One, genera una requisición de compra interna.

A continuación, el área encargada de trabajar con los materiales de la empresa, llamada “Materials Management” utiliza esa información para cargar al sistema SAP los materiales requeridos, por supuesto con las cantidades, y con ello generar una orden de compra.

Enseguida de eso, el propio equipo de Materiales se encarga de enviar a los proveedores las órdenes de compra, esto lo realiza mediante otro sistema de comunicación entre los proveedores y la empresa o directamente mediante un correo.

Algunos días después, el proveedor envía a la empresa otro correo donde se incluye la factura de compra, en el que el proveedor coloca el material que va a enviar, con las cantidades específicas y con la información de a cuál orden de compra se envía cada producto (es importante mencionar esto, debido a que los proveedores no siempre tienen completa disponibilidad de materiales, y por ello envían en una factura materiales de distintas órdenes de compra).

Continuando con el proceso, aquí es donde comienza la parte tediosa para el trabajador del equipo de Materiales, ya que necesita acceder al sistema SAP. Dentro del sistema SAP existen muchas subdivisiones dependiendo el área que se esté trabajando, en este caso se necesita acceder a una división llamada “EP1”, que es un área utilizada por el departamento de Cadena de Suministro, y con ello Demand y Materials Management.

Dentro de esa sección, el trabajador requiere acceder a una transacción llamada “MD04”, dentro de esta transacción el trabajador compara los materiales. Esto sucede ya que para un solo material se manejan 3 números distintos de identificación, el primero es el número de catálogo, que es un número que permite a los proveedores conocer los materiales, el segundo es el Numero de SAP, que es un numero interno de Schaeffler Automotive Aftermarket México para identificar cada material individualmente, y el tercero es el número del proveedor, que es su propio número de identificación.

Y en las facturas enviadas por los proveedores, estos mismos envían cualquiera de los 3 números, por lo que el trabajador se enfrenta al primer gran reto de esta tarea, el obtener el número de SAP, que puede ser realizado de dos formas, la primera es accediendo a la transacción MD04, donde se tiene que colocar el número que haya mandado el proveedor en una casilla indicada y distinta para cada número, y de ahí obtener el número de SAP, esto lo tiene que realizar para cada uno de los números que haya mandado el proveedor en la factura, lo que puede convertirse en una tarea casi imposible o demasiado compleja cuando se trata con facturas con más 50 números distintos (lo que es demasiado común, y se puede ver fácil, sabiendo que hay facturas con más de 200 números distintos). La segunda manera de realizar esto, es accediendo a una base de datos alojada en un documento Excel en el OneDrive de la empresa, donde se tienen los números, y se actualizan cada cierto tiempo. Este proceso no es para nada más simple, ya que necesitan hacer lo mismo, copiar y pegar el número de material proveniente de la factura, y buscarlo en la base de datos en Excel (este proceso es muy lento y se ve afectado muchas veces por la capacidad de procesamiento de la computadora utilizada).

Una vez que se tiene completo el comparativo entre los números, y se ha obtenido el número de SAP del material, el trabajador se apoya de un formato en Excel, donde coloca toda la información requerida por la transacción, como el Numero de Proveedor (que al igual que los números tiene que ser obtenido de la base de datos), el numero de la factura, la fecha de entrega, el Purchase Org, que es siempre una constante, al igual que la Planta. Coloca en una tabla la cantidad del material, el número de material que se ha obtenido de la factura, el Numero de SAP que se ha obtenido anteriormente, y la orden de compra a la que se referencia ese material.

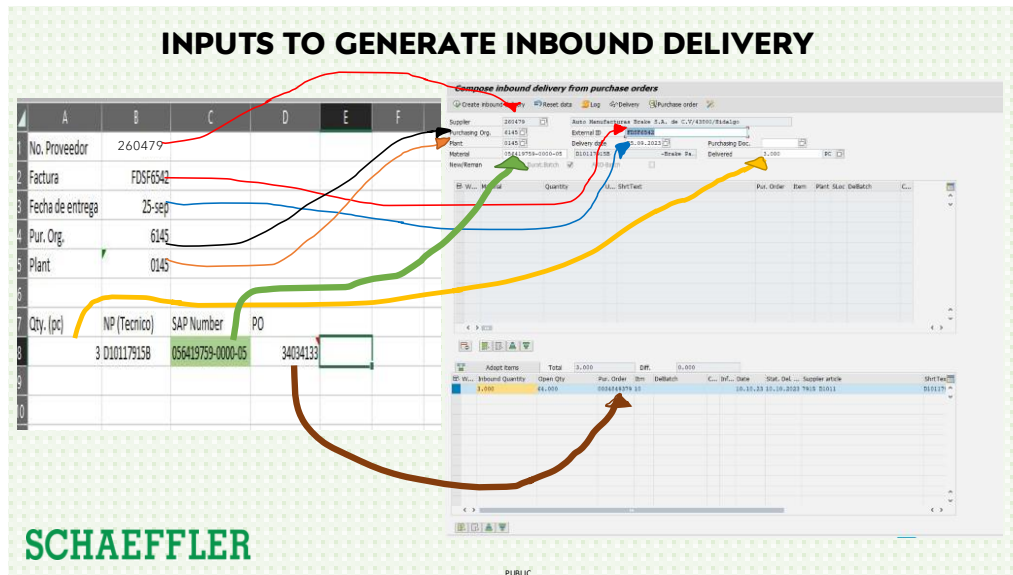


Figura 1: Inputs Inbound

En la Figura 1, se puede observar la transacción tal cual, en SAP, dicha transacción se llama “/SGD/MM_IBDLV_COMP”, y en la parte izquierda de la imagen se observa el formato utilizado por el equipo de materiales para facilitar un poco la tarea.

En la transacción de SAP, el trabajador requiere llenar cada una de las casillas mostradas para cada uno de los materiales a registrar, en la tabla inferior de la imagen anterior se observa una lista con los materiales disponibles, es ahí donde el trabajador requiere analizar si algún número de Orden de compra coincide con el enviado por el proveedor, para entonces registrar el artículo, y que este se muestre en la tabla superior, esto necesita hacerlo manualmente para cada uno de los materiales, y después de ello cuando todos los materiales han sido registrados, debe presionar el botón “Créate Inbound Delivery”, y con ello la Inbound se crea y se genera un número de identificación para la misma.

Una vez que se tiene este número de identificación se requiere que el trabajador lo ingrese a otro archivo en Excel para registrar que el Inbound se ha creado. Aquí es donde termina el proceso del equipo de Materials Management. Y comienza el proceso para la parte del almacén, donde un trabajador de esa área, consulta ese número, y en su área de trabajo una vez que el embarque ha llegado en físico al almacén, compara los materiales registrados en el Inbound con los materiales que realmente llegaron, registra las cantidades y que el material sea el correcto en otra transacción de SAP llamada “SAP SCWM/PRDI”, finalizada esta labor, el sistema crea una etiqueta para indicar el pallet o tarima al cual fue asignado el material entrante, finalizado todo este proceso, el Inbound se marcara como finalizado y todo el proceso será repetido varias veces a la semana para todo el material entrante.

Como una forma de comprender todas las entradas y salidas del proceso, se desarrolló la siguiente tabla, que muestra los Inputs de la transacción en SAP, con su correspondiente fuente de información.

Nombre	Descripción	Fuente
Supplier	El número específico del proveedor.	Factura de Proveedor (nombre del proveedor o pieza que vende) - Transacción MD04 (Se ingresa el número de pieza y esto devuelve el número de proveedor)
Purch Org.	Organización de compras.	Es una constante (6145)
Plant	Planta	Es una constante (0145)
Material	Especifica el número de la pieza o material.	Se obtiene de la factura de compra (es posible ingresar el número de pieza del proveedor), o de la transacción MD04 (número de pieza en el sistema SAP)
External ID	Número de factura	Se obtiene directamente de la factura de compra.
Delivery Date	Es la fecha estimada de entrega para que el paquete llegue a destino.	Se calcula según la fecha de la factura de compra, se suman algunos días dependiendo de varios factores.
Delivered	Es el número de piezas enviadas por el proveedor.	Se obtiene de la factura de compra.
Pur. Order	Es la orden de compra a la que se envía una pieza específica. Puede haber varias órdenes de compra en una factura para un número de pieza. Nota: No es un insumo como tal, se utiliza para comparar y decidir a qué orden de compra se debe declarar el ingreso de material.	Se obtiene de la factura de compra.

Table 1. Entradas de la transacción

Una vez que se ha comprendido completamente el proceso es necesario reconocer algunos problemas comunes a los que se enfrentan los trabajadores al realizar este proceso, esto para poder contemplarlos en las soluciones a desarrollar.

- El proveedor envía menos material del solicitado en la orden de compra.
Esto ocurre cuando el proveedor no tiene la disponibilidad en cantidad completa de algún material solicitado en la orden de compra, cuando esto ocurre el trabajador no requiere modificar nada, solo queda cantidad pendiente para cierto material, que será enviado en el futuro por el proveedor.
- El proveedor envía más cantidad de un material.
Esto ocurre cuando el proveedor se equivoca y envía mayor cantidad de algún material, lo que el trabajador hace es modificar manualmente la cantidad dentro de la transacción en SAP.
- El material viene referenciado a una orden de compra errónea o no existente.
Cuando esto ocurre el trabajador requiere “saltar” u omitir este artículo dentro de la Inbound, para solicitar una aclaración al proveedor y modificar la orden de compra a la correcta.
- Material asignado actualmente a otro Proveedor.
Esto ocurre ya que dentro de la empresa en ocasiones suelen ocurrir modificaciones, es decir, que en algún momento se puede decidir por conveniencia o por alguna otra razón, que cierto material se le compre a algún proveedor distinto, cuando esto ocurre, el trabajador requiere tener en cuenta que proveedor envió el material para registrarlo ante ese proveedor y no ante otro.
- Material Descontinuado.
Dentro de Schaeffler AAM, ciertos materiales suelen ser descontinuados, cuando se sube el material la transacción en SAP, la propia transacción envía un error e impide que se cargue dicho material, cuando sucede esto, el trabajador debe “saltar” u omitir este material, y primeramente solicitar una aclaración al proveedor, y si se desea recibir, se debe solicitar a otro departamento dentro de la empresa que habilite nuevamente dicho material.

Estos son los principales inconvenientes que se pueden suscitar en el proceso, es necesario conocerlos y saber cuál es la forma correcta de tratarlos o evitarlos dentro de la propuesta de automatización.

Ya con toda la información obtenida sobre el proceso a automatizar, se requiere identificar las posibles propuestas de solución.

Para abordar esto se ha decidido tomar 3 posibles formas de hacerlo, ya sea de manera individual o colaborativa, la primera forma es hacer uso de la herramienta de Mendix.

Para analizar el uso de esta herramienta se ha decidido desarrollar una tabla comparativa entre las ventajas y desventajas de su uso.

Mendix	
Ventajas	Desventajas
Desarrollo rápido: Mendix es conocido por su rápido desarrollo de aplicaciones, lo que puede acelerar la implementación de la solución. Interfaz de usuario intuitiva: Mendix permite la creación de interfaces intuitivas y fáciles de usar para los usuarios finales. Colaboración en equipo: facilita la colaboración entre equipos de desarrollo y usuarios comerciales para definir y perfeccionar procesos.	Dependencia de terceros: la integración con SAP ERP y otros sistemas puede requerir asistencia de consultoría externa, lo que agrega complejidad. Personal especializado: Es posible que se requiera personal capacitado de Mendix para el desarrollo de aplicaciones que interactúen con otras bases de datos. Limitaciones de personalización: Mendix puede tener limitaciones a la hora de personalizar ciertos procesos o flujos de trabajo complejos. Dificultad para capturar datos: el usuario tendría que ingresar manualmente cada número de los datos de la factura dentro de la aplicación. Permisos: Se requieren permisos para utilizar las funciones que ofrece este sistema.

Table 2. Ventajas y Desventajas de Mendix

También se analizó de la misma manera la opción de Power Automate.

Power Automate	
Ventajas	Desventajas
Integración con SAP ERP: Power Automate puede integrarse con SAP ERP a través de conectores y adaptadores, lo que facilita la automatización de flujos de trabajo que involucran datos y procesos de SAP. Interfaz de usuario intuitiva: la interfaz de usuario de Power Automate es intuitiva y no requiere conocimientos de programación avanzados, lo que permite a los usuarios empresariales crear flujos de trabajo. Conectividad Versátil: Permite la integración con otras aplicaciones y servicios, facilitando la automatización de procesos que involucran múltiples sistemas junto con SAP ERP, como la conectividad con aplicaciones de Office, como Excel.	Utiliza mucha RAM y recursos de procesamiento del procesador. Limitaciones de personalización: es posible que algunos procesos altamente personalizados en SAP ERP no se puedan automatizar completamente a través de Power Automate debido a limitaciones de personalización. Permisos: Se requieren permisos para utilizar las funciones que ofrece este sistema.

Table 3. Ventajas y Desventajas Power Automate

De la misma manera se realizó un completo análisis de la opción de SAP GUI Scripting API, basada en Visual Basic for Applications.

SAP GUI Scripting API	
Ventajas	Desventajas
Integración profunda con SAP ERP: SAP GUI Scripting API proporciona integración directa con los sistemas SAP ERP, lo que permite un control preciso y la automatización de procesos específicos de SAP. Personalización y control: ofrece un alto nivel de personalización y control sobre los procesos de SAP, lo que lo hace adecuado para flujos de trabajo complejos y altamente personalizados. Sin costos de licencia adicionales: si ya tiene licencias de SAP GUI, generalmente no hay costos de licencia adicionales asociados con el uso de la API de scripting de SAP GUI. Robusto y confiable: Es una forma sólida y confiable de automatizar los procesos de SAP, garantizando coherencia y precisión. Conectividad: Al estar basado en VBA, tiene conectividad directa a múltiples aplicaciones de Microsoft, como Office, incluido "Excel"	Experiencia técnica requerida: el desarrollo y mantenimiento de scripts utilizando la API de scripting GUI de SAP generalmente requiere experiencia técnica y conocimientos de programación con Visual Basic para aplicaciones. Permisos: Se requieren permisos para utilizar las funciones que ofrece este sistema.

Table 4. Ventajas y Desventajas SAP Scripts

Teniendo en cuenta toda la información y el análisis sobre las opciones que se han propuesto se realizó un diagrama para ilustrar cada una de las opciones y la forma en que se podrían llevar a cabo.

A continuación, en la Figura 10, se puede analizar cada uno de los procesos.

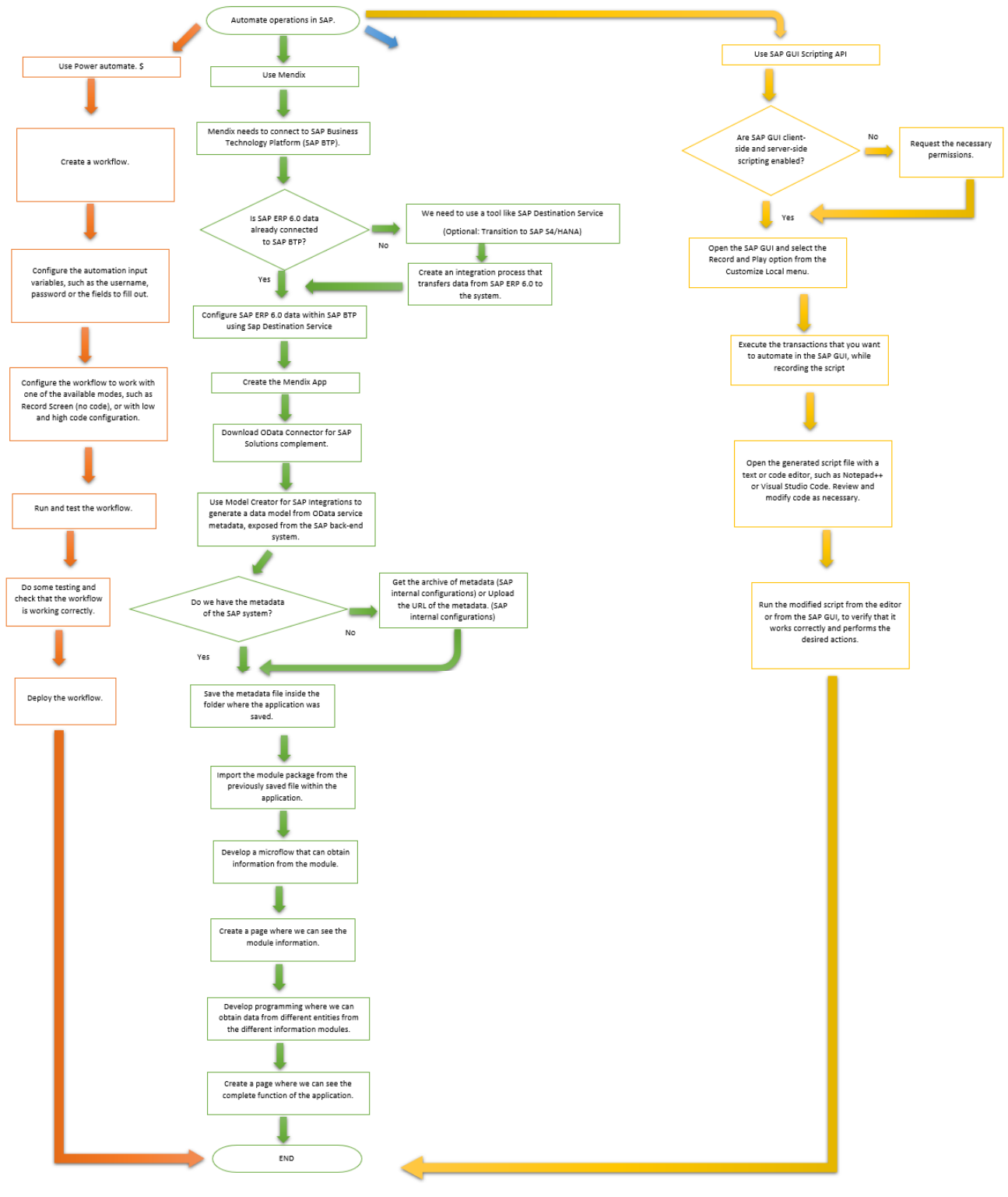


Figura 10. Automatizar Operaciones en SAP

Como se puede ver, el implementar la automatización utilizando la herramienta de Mendix es un gran reto, ya que de todas las opciones propuestas este método es el más extenso y es el método que requiere más condiciones extra para poder llevarse a cabo. Se requiere de software adicional llamado “SAP Business Technology Platform” para permitir una relación entre Mendix y la base de datos del software de SAP ERP 6.0 utilizado por la empresa, y este software solo puede ser implementado por personal externo a la empresa, lo que es un caso no presupuestado para esta solución, por lo cual esta solución para este caso práctico será descartada.

Al analizar la propuesta de utilizar Power Automate para implementar la automatización se puede ver que es una forma mucho menos compleja, ya que permite una conexión más directa con el sistema de SAP y con distintas aplicaciones de Microsoft, como Excel, aunque, una de las desventajas de utilizar esta herramienta es que la misma utiliza algunos recursos computacionales; pero no a un grado que pueda llegar a afectar la experiencia de usuario. Por lo que esta opción es una opción viable para realizar la tarea, con la única condición que requiere que se le dediquen ciertos recursos computacionales para su ejecución.

Finalmente, al analizar la tercera opción “SAP GUI Scripting API”, se puede ver que es una opción viable, puesto que como se ha mencionado anteriormente, es una herramienta nativa del sistema SAP ERP 6.0, y adicionalmente se basa en el lenguaje por excelencia de Microsoft, “Visual Basic for Applications”, este lenguaje tiene una peculiaridad muy grande, y es que es el lenguaje que manejan muchas de las aplicaciones de Microsoft, como lo son las aplicaciones de Microsoft Office, como Word o Excel, lo cual viene a ser una gran noticia, puesto que esta última puede ser utilizada como plantilla para facilitar la automatización. Aunque tiene la gran desventaja, que es una herramienta muchas veces deshabilitada por las organizaciones, debido a que existe un riesgo de que al utilizarla se ejecuten acciones no permitidas.

Por lo que finalmente, después de consultarlo con distintas áreas de la empresa, se decidió que se utilizarían dos de las propuestas, en primera parte Visual Basic for Applications, para trabajar directamente con Excel y automatizar la tarea de la obtención de los números de SAP, y en segunda parte con Power Automate, para trabajar como una RPA (Robotic Process Automation), y realizar de forma automática la carga de toda la información dentro de la transacción de SAP, además de servir como conexión entre este sistema y el propio libro de Excel que se vaya a utilizar.

Como primera parte del proceso se procedió a desarrollar la interfaz de lo que sería la aplicación, para realizar esta misma se contempló toda la información que es requerida en la transacción de SAP y que es obtenida directamente de la factura recibida de parte del proveedor.

Como se observa en la Figura 11, la información de la factura es obtenida de un documento. Por cuestiones de privacidad se han ocultado datos relevantes.

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Domicilio fiscal TIZAYUCA, HIDALGO, MEXICO R.F.C. : Regimen Fiscal: (601)General de Ley Personas Morales Expedido en TIZAYUCA,		FACTURA 857 AOC1414A-AE20-4A88-8911-044C4073DD2A FECHA 2023-10-03T09:16:56				
Comprobante fiscal digital (l)Ingreso		T/C :				
DEBO Y PAGARE INCONDICIONALMENTE A LA ORDEN DE AUTO MANUFACTURAS BRAKE SA DE CV LA CANTIDAD CONSIGNADA EN ESTE TITULO. EL PRESENTE PAGARE REPRESENTA LA ACEPTACION DE LOS TERMINOS COMERCIALES QUE FIJA AUTO MANUFACTURAS BRAKE SA DE CV. SI ESTE PAGARE NO ES LIQUIDADO EN SU VENCIMIENTO, CAUSARA UN INTERES MORATORIO DEL 3% MENSUAL, ADEMAS DE LOS GASTOS DE COBRO						
FACTURA A : RFC: TH8 Uso de CFDI: (G01)Adquisición de mercancías. No.Cte 239		Vendedor:				
SCHAEFFLER AUTOMOTIVE AFTERMARKET MEXICO		Condiciones: 2o. DÍA DEL SIGUIENTE MES				
Régimen Fiscal: (601)General de Ley Personas Morales		Metodo de Pago y Cta				
Calle: AVENIDA QUETZAL, MEXICO, Teléfono:55 11		(PPD)Pago en parcialidades o diferido (99)Por definir				
No. de Prov:		No. de Procedimiento:				
O. de Compra:						
CANTIDAD	U / SAT	PROD/SAT	DESCRIPCION	DESC	P / UNIT	TOTAL
140.00	pz H87	D1801	8 BALATA SEMMETALICA	0.00	7.4800	1,047.20
360.00	pz H87	D11918	8 BALATA SEMMETALICA	0.00	8.0500	2,898.00
179.00	pz H87	CMD143	8 BALATA CERAMICA	0.00	9.2200	1,650.38
284.00	pz H87	D11918	8 BALATA SEMMETALICA	0.00	8.0500	2,286.20
120.00	pz H87	CMD768763508	8 BALATA CERAMICA	0.00	9.7600	1,171.20
119.00	pz H87	CMD34	8 BALATA CERAMICA	0.00	6.5200	775.88
19.00	pz H87	CMD17	8 BALATA CERAMICA	0.00	16.3300	310.27
59.00	pz H87	CMD202492548	8 BALATA CERAMICA	0.00	8.6200	508.58
64.00	pz H87	D101	8 BALATA SEMMETALICA	0.00	13.3800	856.32
32.00	pz H87	D108	8 BALATA SEMMETALICA	0.00	12.6500	404.80
80.00	pz H87	D180	8 BALATA SEMMETALICA	0.00	7.3000	584.00

Figura 11. Ejemplo de Factura

Se decidió que la mejor manera de abordar la interfaz era utilizando una plantilla en la aplicación de Microsoft Excel, esto se decidió al analizar que con esta aplicación ya existe una gran facilidad para procesar grandes cantidades de filas y columnas, y como se observó las facturas recibidas del proveedor tienen una cantidad bastante grande de estas mismas. Por lo que no resultaría viable desarrollar otra interfaz que maneje de esta manera la información, solo sería una pérdida de tiempo.

Por lo cual, se desarrolló la interfaz mostrada en la Figura 12:

Automation of the generation of Inbound Numbers					
Supplier	Supplier Name				
External ID (Invoice Number)	Inbound Number				
Delivery Date	Inbound Number				
Pur. Org.	Inbound Number				
Plant	Inbound Number				

Delivered (Qty)	Description (Catalog Number)	Purchase Order	SAP Number	Material STATUS	STATUS

Figura 12. Interfaz 1

Como se puede observar en la Figura 12, en esta interfaz se incluyeron las casillas que permiten extraer la información de la factura, se observa que en la parte superior se incluyen las casillas que permiten colocar el nombre y el número del proveedor. En las siguientes filas se colocaron las casillas que permiten colocar el número de factura, la fecha de esta, y las constantes como lo son la planta y la organización de compras.

A continuación, se observa que se incluyó una tabla, en la cual se plantea colocar los materiales contenidos en la factura, en la primera columna se propone colocar la cantidad, en la segunda columna se propone colocar el número de material, este número de material es el llamado “Numero de Catalogo”, que es un numero en común que se comparte con los proveedores para identificar los productos. (Este número no es funcional ni detectado por la transacción en SAP, por lo que es necesario convertirlo al número utilizado internamente por la empresa, el llamado “Numero de SAP”), en la siguiente columna se tiene la columna donde se debe colocar la orden de compra a la cual fue enviado el material, esto sabiendo que en una factura normalmente los proveedores envían materiales que están referenciados a distintas órdenes de compra. La siguiente columna es la llamada “SAP Number”, en esta columna se planea que sea colocado el número de SAP específico para cada material.

En la siguiente columna se tiene la llamada “Material_STATUS”, que es una columna informativa diseñada para tratar uno de los posibles errores en este proceso, donde se mostrara la información del material, sabiendo que internamente dentro de Schaeffler cada material tiene un estatus, este estatus define como se encuentra el material, si tiene el código “61”, se considera que este material se encuentra en un estado normal, y esto quiere decir que puede ser utilizado y contemplado dentro de las transacciones en SAP, si tiene algún otro número esto significa que el material o no se encuentra disponible, o esta discontinuado o en general ocurre algo que no permite que pueda ser utilizado dentro de la empresa, por lo que al llenar esta columna, de forma sencilla es posible evitar que la transacción marque un error al intentar subir un material no disponible.

La última columna es otra columna informativa, donde se planea mostrar si se realizó correctamente la carga de la información, o si hubo algún error que impidió que se llevara a cabo correctamente.

Después de realizar esta interfaz, se procedió a analizar cómo realizar de forma automática la comparativa para la obtención de los Números de SAP para los materiales.

Analizando el proceso tradicional para la obtención de estos números, se contempló que se realizaba obteniendo la información de una base de datos alojada en otro documento de Excel, esta base de datos se actualiza cada cierto tiempo y tiene un registro donde se compara el número de SAP y el número de catálogo, y tiene también el registro del estatus del material.

Esta búsqueda y obtención de datos era realizada utilizando otro documento donde se accedía número por número con la opción propia de la aplicación llamada “Buscar”, que es por cierto un proceso demasiado desgastante, considerando que hay facturas con más de 200 números de material distinto.

Por lo cual se decidió aprovechar esta base de datos, y aprovechar aún más que esta alojada en otro documento de Microsoft Excel, por lo que es la opción perfecta para la utilización de Visual Basic for Applications. Así que se comenzó a desarrollar la estrategia de programación para realizar todas estas consultas de forma automática.

Primeramente, al estar trabajando con este lenguaje, es necesario tener un orden en el código de programación, por lo que se decidió trabajar con Módulos Individuales, dentro del módulo es necesario siempre colocar la programación requerida.

Así que se comenzó a desarrollar el bloque de programación, como en todo lenguaje es necesario comenzar con las variables.

```
Sub Generar_SAPNumber()  
    ' Declaracion de variables  
    Dim wsGenSAPNumber As Worksheet  
    Dim ws0145_Active As Worksheet  
    Dim tablaSAPNum As ListObject  
    Dim tablaItemActive As ListObject  
    Dim SuppNum As Range  
    Dim Description_Number As Range  
    Dim MaterialCell As Range  
    Dim DescriptionCell As Range  
    Dim SupplierNum_ItemActive As Range  
  
    ' Declaracion de variables para las columnas  
    Dim CatalogNumber_Column As ListColumn  
    Dim DescriptionColumn As ListColumn  
    Dim SupplierColumn As ListColumn  
    Dim SupNumberColumn As ListColumn  
    Dim MaterialValue As Variant
```

Figura 13. Variables Programa

Como se puede observar en la Figura 13, se deben asignar las variables a las hojas de trabajo para referirse a ellas de forma simple, también se deben declarar las variables para referirse a las tablas, a las columnas de las propias tablas, y a los rangos, que son espacios dentro de las hojas a los que se desea acceder.

A continuación, es necesario asignar objetos reales a esas variables, pero hay que considerar que se estará trabajando con dos documentos distintos, por lo que es necesario establecer una estrategia que permita que el programa funcione correctamente ante cualquier error, como que no está abierto el documento de la fuente de datos, o que tenga un nombre distinto, por lo que la siguiente secuencia de código permite trabajar con ello.

Primeramente, se pedirá al usuario que ingrese el nombre del archivo que funciona como base de datos, y se añadirán condicionales para el manejo de los errores, quedando de la manera mostrada en la Figura 14.

```
' Referenciamos las hojas y las tablas
Set wsGensAPNumber = ThisWorkbook.Sheets("Hoja_GensAPNumber_MMav")
' Pedir al usuario el nombre del archivo
Dim fileName As String
fileName = InputBox("Por favor, ingrese el nombre del archivo (sin incluir la extensión):", "Nombre del archivo Excel")
' Verificar si se ingresó un nombre de archivo
If fileName <> "" Then
    ' Concatenar la extensión del archivo al nombre ingresado por el usuario
    fileName = fileName & ".xlsx"
    ' Referenciar el libro de trabajo usando el nombre ingresado por el usuario
    Dim targetWorkbook As Workbook
    On Error Resume Next ' Evitar errores si el archivo no se encuentra
    Set targetWorkbook = Workbooks(fileName)
    On Error GoTo 0 ' Restablecer el manejo de errores
    ' Verificar si se encontró el libro de trabajo
    If Not targetWorkbook Is Nothing Then
        ' Referenciar la hoja usando el nombre ingresado por el usuario
        Set ws0145_Active = targetWorkbook.Sheets("0145_Active")
    End If
End If
```

Figura 14: Pedir nombre de archivo

Como se pudo analizar en la Figura 14, primeramente se referencio la hoja en la que se ejecutara el código, con el comando ThisWorkbook.Sheets(), en seguida mediante un input box se solicita al usuario que ingrese el nombre del archivo sin incluir su extensión, y en seguida se comienza a realizar el manejo de los errores, para que el programa continúe funcionando si el usuario no ingresa ningún nombre de archivo, o si ingresa un nombre incorrecto, incluso se contempla que el usuario ingrese el nombre de otro archivo abierto, pero si en ese archivo no se encuentra la hoja indicada en el código, se le mandara al usuario una advertencia indicándole que no se ha encontrado el libro.

El inputbox se muestra en la Figura 15:

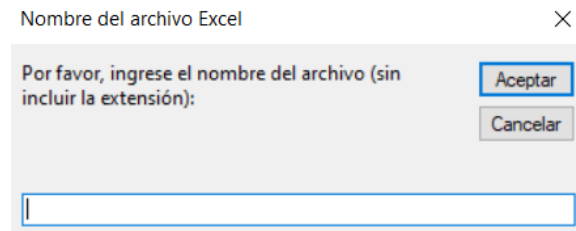


Figura 15: InputBox

Y si el usuario se equivoca o no introduce el nombre, se le enviara la advertencia mostrada en la Figura 16:

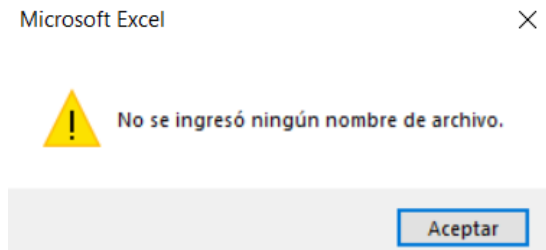


Figura 16: Error con el nombre de archivo

A continuación, en el código, se comienza a trabajar con las variables declaradas al inicio, se comienza a asignarles columnas específicas de los libros abiertos.

```
' Si se ha abierto el archivo se comienza a trabajar
Set tablaSAPNum = wsGenSAPNumber.ListObjects("Tabla_SAPNum")
Set tablaItemActive = ws0145_Active.ListObjects("Item_0145_Active")

' Columnas tabla "Tabla_SAPNum"
Set CatalogNumber_Column = tablaSAPNum.ListColumns("Description (Catalog Number)")
Set Sup_ItemNumber = tablaSAPNum.ListColumns("Supplier-item number")

' Columnas tabla "Item_0145_Active"
Set DescriptionColumn = tablaItemActive.ListColumns("Description")
Set SupplierColumn = tablaItemActive.ListColumns("Supplier")
Set SupNumberColumn = tablaItemActive.ListColumns("Sup Number")
Set Sup_Item_Number = tablaItemActive.ListColumns("Supplier-item number")
```

Figura 17: Asignación de columnas

En seguida, se comienza a manejar otro posible error que podría ocurrir, y este error es que el usuario presione el botón, introduzca el nombre del archivo, pero la tabla de información se encuentre vacía, y si esto ocurre el código se interrumpirá y enviara un desagradable error al usuario, por lo que se utilizó el condicional mostrado en la Figura 18, para evitar este error.

```
If tablaSAPNum.ListRows.Count > 0 Then ' Verificar si la tabla tiene filas de datos

Else
    ' Mostrar mensaje si la tabla está vacía
    MsgBox "La tabla está vacía.", vbInformation
End If
```

Figura 18: Manejo de error tabla vacía

Cuando la tabla tiene datos, se comienza a trabajar para obtener la información deseada, para hacer esto se decidió que la manera más eficiente, hablando de capacidad de procesamiento, es utilizar bucles “For”, con estos bucles se comenzara a acceder a los datos contenidos en las columnas anteriormente declaradas y se asignara su valor en las variables mostradas:

```
'Obtener SAP Number, Supplier - Item Number y Material STATUS
' Iteramos sobre cada fila en la columna Description (Catalog Number)
For Each SuppNum In CatalogNumber_Column.DataBodyRange
    ' Iteramos sobre cada fila en la tabla de Item Active, columna Description
    For Each Description_Number In DescriptionColumn.DataBodyRange
```

Figura 19: Ciclo For

Ya que se ha accedido a la información contenida en las columnas especificadas, se necesita hacer una comparación, para identificar cuando un numero coincide, pero es necesario considerar la naturaleza de esos números, y resulta obligatorio convertir los valores a texto antes de ser comparados, para esto se utiliza el comando “CStr”:

```
' Comparamos Catalog Number con Description
If CStr(SuppNum.Value) = CStr(Description_Number.Value) Then ' Convierte los valores a texto antes de comparar
```

Figura 20: Comparación entre valores

Una vez que se ha realizado la comparativa, y que, por supuesto se ha encontrado la coincidencia, se comenzara a asignar el valor correspondiente para ese número en las columnas mostradas.

Para hacer esta asignación de valores, se utilizó el comando “Offset”, que es un comando que se encarga de “moverse” en las columnas y en las filas a partir de una posición de origen especificada, su formato es el siguiente: Offset(0, 2).Value. Donde los valores entre paréntesis indican la posición, en este caso el 0 indica que no se mueve de fila, y el 2 indica que se moverá 2 columnas a la derecha (contemplando que también se pueden utilizar numerosa negativos para moverse en la dirección contraria). El código se muestra en la Figura 21.

```
SuppNum.Offset(0, 2).Value = Description_Number.Offset(0, -1).Value
' Colocamos el valor de SAP-Status en la columna Material_STATUS
SuppNum.Offset(0, 4).Value = Description_Number.Offset(0, 7).Value
' Colocamos el numero de Material del proveedor
SuppNum.Offset(0, 3).Value = Description_Number.Offset(0, 4).Value
```

Figura 21: Asignación de valores

A continuación, para efectos de simplicidad, se decidió mostrar al usuario cuando un material maneja un estatus distinto del antes mencionado “61”, para esto se utilizó un condicional, que va a colorear la casilla con ese número de un color distintivo.

```
' Comparar Material_STATUS con "61" y colorear la celda de Material_STATUS si es diferente de "61"
If SuppNum.Offset(0, 4).Value <> "61" Then ' SuppNum.Offset(0, 4) se refiere a la columna Material_STATUS
    SuppNum.Offset(0, 4).Interior.Color = RGB(255, 0, 0) ' Relleno rojo para la celda Material_STATUS
End If
```

Figura 22: Colorear Status

El código completo de este segmento luce como se observa en la Figura 23:

```
'Obtener SAP Number, Supplier - Item Number y Material_STATUS
' Iteramos sobre cada fila en la columna Description (Catalog Number)
For Each SuppNum In CatalogNumber_Column.DataBodyRange
    ' Iteramos sobre cada fila en la tabla de Item Active, columna Description
    For Each Description_Number In DescriptionColumn.DataBodyRange
        ' Comparamos Catalog Number con Description
        If CStr(SuppNum.Value) = CStr(Description_Number.Value) Then ' Convierte los valores a texto antes de comparar
            ' Encontramos una coincidencia, colocamos el valor de la fila correspondiente en SAP Number
            SuppNum.Offset(0, 2).Value = Description_Number.Offset(0, -1).Value
            ' Colocamos el valor de SAP-Status en la columna Material_STATUS
            SuppNum.Offset(0, 4).Value = Description_Number.Offset(0, 7).Value
            ' Colocamos el numero de Material del proveedor
            SuppNum.Offset(0, 3).Value = Description_Number.Offset(0, 4).Value

            ' Comparar Material_STATUS con "61" y colorear la celda de Material_STATUS si es diferente de "61"
            If SuppNum.Offset(0, 4).Value <> "61" Then ' SuppNum.Offset(0, 4) se refiere a la columna Material_STATUS
                SuppNum.Offset(0, 4).Interior.Color = RGB(255, 0, 0) ' Relleno rojo para la celda Material_STATUS
            End If
        End If
    Next Description_Number
End If
Next SuppNum
```

Figura 23: Obtención de valores

Para continuar, se necesita encontrar el nombre de proveedor, y por supuesto su número, ya que es un dato requerido dentro de la transacción de SAP, para ello se hará uso de un

método similar de búsqueda, donde se va a comparar el número de material de ambos libros, cuando se encuentre una coincidencia se va a colocar el número y el nombre en las casillas indicadas.

```
' Asignar Supplier y Supplier Number
' Obtenemos el valor de la primera fila de Description (Catalog Number)
MaterialValue = CStr(CatalogNumber_Column.DataBodyRange(1, 1).Value)
' Iteramos sobre cada fila en la tabla de Item Active
For Each Item In DescriptionColumn.DataBodyRange
    ' Comparamos Material con Description
    If MaterialValue = Item.Value Then
        ' Encontramos una coincidencia, colocamos el valor de Supplier en N2 de Hoja_GenSAPNumber_MMav
        wsGenSAPNumber.Range("N2").Value = SupplierColumn.DataBodyRange(Item.Row)
        ' Colocamos el valor de Sup Number en L2 de Hoja_GenSAPNumber_MMav
        wsGenSAPNumber.Range("L2").Value = SupNumberColumn.DataBodyRange(Item.Row)
        ' Salimos del bucle ya que encontramos una coincidencia
    Exit For
    End If
Next Item
```

Figura 24: Asignación de Proveedor

Como se observa en la Figura 24, se utilizan algunos métodos distintos, como el “DataBodyRange”, que devuelve un objeto Range que representa el intervalo de valores, excepto la fila de encabezado, en una tabla. O el “Range” que se utiliza para referirse a un rango específico dentro de la hoja. Estos métodos se utilizan, ya que, en la tabla de destino, solo deseamos colocar un número, y no colocar todas las coincidencias de datos para todos los materiales.

Para complementar el programa, se decidió añadir una funcionalidad extra, y esta se basa en que algunas veces los proveedores envían las facturas con un número distinto al número de catálogo, este número es el llamado número de proveedor directamente, y se busca que se puedan automatizar todas las tareas, por lo que se desarrolló otra programación que permite que el usuario pueda ingresar cualquiera de los dos números, y de manera automática se arrojen el resto de los datos. Así que la vista general del programa es la mostrada en la Figura 25:

Automation of the generation of Inbound Numbers (MM Availability)						
Supplier		Supplier Name				
External ID (Invoice Number)						
Delivery Date		Inbound Number				
Pur. Org						
Plant						
				Clean All		
Delivered (Qty)	Description (Catalog Number)	Purchase Order	SAP Number	Supplier-item number	Material STATUS	STATUS

Figura 25: Interfaz del programa 2

Para abordar esto, primeramente, se debe verificar cuál de las columnas tiene datos, por lo cual se hará uso de un método nuevo, “ListRows.Count”, que consiste en acceder a las filas de la tabla, y mediante “Count”, se realiza el conteo de cuantas filas activas existen en la tabla.

El código es el mostrado en la Figura 26.

```
If Application.WorksheetFunction.CountA(CatalogNumber_Column.DataBodyRange) > 0 Then '
    'Obtener SAP Number, Supplier - Item Number y Material_STATUS

Else
    ' La columna "Description (Catalog Number)" está vacía, Supplier-item number tiene datos

Else
    ' Mostrar mensaje si la tabla está vacía
    MsgBox "No hay datos en las columnas Description ni Supplier-item number.", vbInformation
End If
```

Figura 26: Comparativa de columnas

Primeramente se analiza si hay filas con datos en la columna “CatalogNumber_Column”, esto se hace mediante el método “CountA”, acompañado de DataBodyRange, y mediante el uso del condicional se evalúa si el número de filas con datos es mayor a 0, para realizar toda la codificación mostrada en la Figura anterior, en caso contrario se hace uso de “Else”, para acceder a la programación adecuada para obtener los datos cuando la columna “Supplier-item number” tiene datos, y finalmente, cuando ninguna de las dos columnas tiene datos, se envía un MsgBox para avisar al usuario que las columnas están vacías.

El código utilizado cuando la columna “Supplier-item number” tiene datos es muy similar al explicado anteriormente, con la variación de que se asignan y comparan distintas columnas. El código es el mostrado en la Figura 27:

```
Else
    ' La columna "Description (Catalog Number)" está vacía, Supplier-item number tiene datos
    If Application.WorksheetFunction.CountA(Sup_ItemNumber.DataBodyRange) > 0 Then
        'Obtener SAP Number, Description (Catalog Number) y Material_STATUS
        ' Iteramos sobre cada fila en la columna Supplier Item Number
        For Each SuppNum In Sup_ItemNumber.DataBodyRange
            ' Iteramos sobre cada fila en la tabla de Item Active en la columna Supplier Item Number
            For Each SupplierNum_ItemActive In Sup_ItemNumber.DataBodyRange
                ' Comparamos Supplier-item number de ambas tablas
                If CStr(SuppNum.Value) = CStr(SupplierNum_ItemActive.Value) Then ' Convierte los valores a texto antes de comparar
                    ' Encontramos una coincidencia, colocamos el valor de la fila correspondiente en SAP Number
                    SuppNum.Offset(0, -1).Value = SupplierNum_ItemActive.Offset(0, -5).Value
                    ' Colocamos el valor de SAP-Status en la Columna Mat_STATUS
                    SuppNum.Offset(0, 1).Value = SupplierNum_ItemActive.Offset(0, 3).Value
                    ' Colocamos el numero de Material del proveedor
                    SuppNum.Offset(0, -3).Value = SupplierNum_ItemActive.Offset(0, -4).Value |
                    ' Comparar Material_STATUS con "61" y colorear la celda de Material_STATUS si es diferente de "61"
                    If SuppNum.Offset(0, 1).Value <> "61" Then ' SuppNum.Offset(0, 3) se refiere a la columna Material_STATUS
                        SuppNum.Offset(0, 1).Interior.Color = RGB(255, 0, 0) ' Relleno rojo para la celda Material_STATUS
                    End If
                End If
            Next SupplierNum_ItemActive
        Next SuppNum

        ' Asignar Supplier y Supplier Number
        ' Obtenemos el valor de la primera fila de Description (Catalog Number)
        MaterialValue = CStr(CatalogNumber_Column.DataBodyRange(1, 1).Value)
        ' Iteramos sobre cada fila en la tabla de Item Active
        For Each Item In DescriptionColumn.DataBodyRange
            ' Comparamos Material con Description
            If MaterialValue = Item.Value Then
                ' Encontramos una coincidencia, colocamos el valor de Supplier en N2 de Hoja_GenSAPNumber_MMav
                wsGenSAPNumber.Range("N2").Value = SupplierColumn.DataBodyRange(Item.Row)
                ' Colocamos el valor de Sup Number en L2 de Hoja_GenSAPNumber_MMav
                wsGenSAPNumber.Range("L2").Value = SupNumberColumn.DataBodyRange(Item.Row)
                ' Salimos del bucle ya que encontramos una coincidencia
            End If
        Next Item
    Else
        ' Mostrar mensaje si la tabla está vacía
        MsgBox "No hay datos en las columnas Description ni Supplier-item number.", vbInformation
    End If
End If
```

Figura 27: Asignación de datos 2 caso

Finalmente, en la última parte del código, se realiza el cierre de todos los ciclos y bucles que se utilizaron, y se finaliza el código en “End Sub”.

```

        End If
    Else
        ' Mostrar mensaje si la tabla está vacía
        MsgBox "No hay datos en las columnas Description ni Supplier-item number.", vbInformation
    End If
End If
Else
    ' Mostrar mensaje si la tabla está vacía
    MsgBox "La tabla está vacía.", vbInformation
End If
' Si no se ha abierto el archivo
Else
    MsgBox "El archivo especificado no se encontró.", vbExclamation
End If
Else
    MsgBox "No se ingresó ningún nombre de archivo.", vbExclamation
End If
MsgBox "Finalizado", vbInformation
End Sub

```

Figura 28: Cierre de Código

A continuación, se decidió que para evitar errores en la transacción, resulta muy útil realizar una comparativa, esta comparativa ocurre cuando los proveedores envían el número de SAP en las facturas (no es una práctica nada común, pero en algunas ocasiones lo realizan de esta manera), cuando esto ocurre es necesario cerciorarse de que han enviado el numero correcto, y para hacer esto se implementó otro botón que es capaz de realizar la comparativa de los números de SAP provenientes de la factura, y los números de SAP de la base de datos.

Para esto se creó otro modulo dentro de Visual Basic. Como en la codificación anterior primeramente se declararon las variables de los objetos a utilizar.

```

Sub Generar_SupplierNumber_MMav()
    ' Declaracion de variables
    Dim wsGenSAPNumber As Worksheet
    Dim ws0145_Active As Worksheet
    Dim tablaSAPNum As ListObject
    Dim tablaItemActive As ListObject
    Dim SAPNum As Range
    Dim ItemNumber As Range
    Dim MaterialCell As Range
    Dim DescriptionCell As Range

    ' Declaracion de variables para las columnas
    Dim MaterialColumn As ListColumn
    Dim SAPNumberColumn As ListColumn
    Dim DescriptionColumn As ListColumn
    Dim Item_NumberColumn As ListColumn

```

Figura 29: Variables proveedor

Como en el caso anterior se requiere solicitar al usuario el nombre del archivo, y seguido de ello asignar objetos reales a las variables, como se muestra en la Figura 30:

```

' Pedir al usuario el nombre del archivo
Dim fileName As String
fileName = InputBox("Por favor, ingrese el nombre del archivo (sin incluir la extensión):", "Nombre del archivo Excel")
' Verificar si se ingresó un nombre de archivo
If fileName <> "" Then
    ' Concatenar la extensión del archivo al nombre ingresado por el usuario
    fileName = fileName & ".xlsx"
    ' Referenciar el libro de trabajo usando el nombre ingresado por el usuario
    Dim targetWorkbook As Workbook
    On Error Resume Next ' Evitar errores si el archivo no se encuentra
    Set targetWorkbook = Workbooks(fileName)
    On Error GoTo 0 ' Restablecer el manejo de errores
    ' Verificar si se encontró el libro de trabajo
    If Not targetWorkbook Is Nothing Then
        ' Referenciar la hoja usando el nombre ingresado por el usuario
        Set ws0145_Active = targetWorkbook.Sheets("0145_Active")
        ' Si se ha abierto el archivo se comienza a trabajar
        Set tablaSAPNum = wsGenSAPNumber.ListObjects("Tabla_SAPNum")
        Set tablaItemActive = ws0145_Active.ListObjects("Item_0145_Active")

        ' Columnas tabla "Tabla_SAPNum"
        Set MaterialColumn = tablaSAPNum.ListColumns("Description (Catalog Number)")
        Set SAPNumberColumn = tablaSAPNum.ListColumns("SAP Number")
        ' Columnas tabla "Item_0145_Active"
        Set DescriptionColumn = tablaItemActive.ListColumns("Description")
        Set Item_NumberColumn = tablaItemActive.ListColumns("Item Number")
        Set SupplierColumn = tablaItemActive.ListColumns("Supplier")
        Set SupNumberColumn = tablaItemActive.ListColumns("Sup Number")
    End If
End If

```

Figura 30: Solicitar nombre de documento

Seguido de ello se requiere cerciorarse que la tabla tenga datos, y una vez que se ha confirmado se comienza a realizar una comparativa parecida a la codificación anterior, por lo que se convierten los valores a texto, se comparan, una vez que se encuentra la conciencia, se asignan los valores correspondientes a la posición adecuada de la tabla.

```

' Verificar si la tabla tiene filas de datos
If tablaSAPNum.ListRows.Count > 0 Then
    tablaSAPNum.ListColumns("Description (Catalog Number)").DataBodyRange.NumberFormat = "@"
    ' Iteramos sobre cada fila en la tabla de SAP Numbers en la columna SAP Number
    For Each SAPNum In SAPNumberColumn.DataBodyRange
        ' Iteramos sobre cada fila en la tabla de Item Active, en la columna Item number
        For Each ItemNumber In Item_NumberColumn.DataBodyRange
            ' Comparamos SAP Number con Item Number
            If CStr(SAPNum.Value) = CStr(ItemNumber.Value) Then 'Se convierte en texto para comparar
                ' Encontramos una coincidencia, colocamos el valor de la fila correspondiente en Description (Catalog Number)
                SAPNum.Offset(0, -2).Value = ItemNumber.Offset(0, 1).Value
                ' Colocamos el valor de SAP-Status en la columna Mat_STATUS
                SAPNum.Offset(0, 2).Value = ItemNumber.Offset(0, 8).Value
                ' Colocamos el numero de Material del proveedor
                SAPNum.Offset(0, 1).Value = ItemNumber.Offset(0, 5).Value
                ' Comparar Material_STATUS con "61" y colorear la celda de Material_STATUS si es diferente de "61"
                If SAPNum.Offset(0, 2).Value <> "61" Then ' SAPNum.Offset(0, 2) se refiere a la columna Material_STATUS
                    SAPNum.Offset(0, 2).Interior.Color = RGB(255, 0, 0) ' Relleno rojo para la celda Material_STATUS
                End If
            End If
            Exit For ' Salimos del bucle interno ya que encontramos una coincidencia
        Next ItemNumber
    Next SAPNum
Else
    ' Mostrar mensaje si la tabla está vacía
    MsgBox "La tabla está vacía.", vbInformation
End If

```

Figura 31: Asignación de valores con SAP Number

Como se pudo analizar, el procedimiento fue similar al utilizado en el otro modulo, con la principal diferencia que se comparan y se asignan los datos a distintas columnas de información de la tabla.

Para continuar con el proceso, se observó que es necesario limpiar las filas de la tabla para poder trabajar con otras facturas, por lo que se implementó el código mostrado en la Figura 32:

```

Sub Clean_MM_Av()
' Borrar datos de las casillas específicas
ThisWorkbook.Sheets("Hoja_GenSAPNumber_MMAv").Range("L2,L3,L4,L5,L6,N2,N4").ClearContents

' Borrar filas, datos y formato de la tabla "Tabla_SAPNum"
Dim wsGenSAPNumber As Worksheet
Dim tablaSAPNum As ListObject

' Referenciar la hoja y la tabla
Set wsGenSAPNumber = ThisWorkbook.Sheets("Hoja_GenSAPNumber_MMAv")
Set tablaSAPNum = wsGenSAPNumber.ListObjects("Tabla_SAPNum")

' Borrar filas y formato de la tabla
If Not tablaSAPNum.DataBodyRange Is Nothing Then
    tablaSAPNum.DataBodyRange.Clear
    tablaSAPNum.DataBodyRange.Delete
End If
End Sub

```

Figura 32: Limpiar tabla

Primero se requiere comparar si la tabla tiene datos, y cuando esto es así, se utiliza el método “Clear”, para borrar todos los datos, y el método “Delete” para eliminar las filas.

Se decidió añadir un instructivo, que permite al usuario manejar el primer parte del programa, en la Figura 33 se puede observar el resultado:

Instrucciones.

1. Antes de abrir este archivo, abrir **MM Availability**.
2. Rellenar las casillas de: External ID (Invoice Number), Delivery Date, Pur. Org., Plant.
3. Rellenar los datos de las columnas : Delivered (Qty), Description (Catalog Number) y Purchase Order (O si se tiene Supplier-item number en vez de Description (Catalog Number)
4. Copiar el nombre completo de **MM Availability**, (Ejemplo: MM Availability_0145_CW40_2023)
5. Presionar el boton "Generate SAP Number"
6. En el mensaje creado pegar el nombre del archivo antes copiado.
- 6.1 Si se tiene el SAP Number, colocarlo en la columna correspondiente "SAP Number", presionar el boton "Generate Catalog Number", y colocar el nombre del archivo.
7. Esperar a que termine el proceso.
8. Abrir Power Automate
9. Seleccionar el flujo "SAP -Inbound Numbers", y darle clic al boton Ejecutar.
10. Esperar a que termine el proceso.
11. Presionar el boton "Clean All" para limpiar todo.

Figura 33: Instrucciones

La interfaz de usuario final quedo como se muestra en la Figura 34:

Instrucciones.

1. Antes de abrir este archivo, abrir MM Availability.
2. Rellenar las casillas de: External ID (Invoice Number), Delivery Date, Pur. Org., Plant.
3. Rellenar los datos de las columnas: Delivered (Qty), Description (Catalog Number) y Purchase Order (O si se tiene Supplier-Item number en vez de Description (Catalog Number)).
4. Copiar el nombre completo de MM Availability (Ejemplo: MM Availability_0145_CW40_2023).
5. Presionar el boton "Generate SAP Number".
6. En el mensaje creado pegar el nombre del archivo antes copiado.
- 6.1 Si se tiene el SAP Number, colocarlo en la columna correspondiente "SAP Number", presionar el boton "Generate Catalog Number", y colocar el nombre del archivo.
7. Esperar a que termine el proceso.
8. Al ser "Power Automate".
9. Seleccionar el tipo "SAP - Inbound Numbers", y darle clic al boton Ejecutar.
10. Esperar a que termine el proceso.
11. Presionar el boton "Clean All" para limpiar todo.

Automation of the generation of Inbound Numbers (MM Availability)

Supplier	Supplier Name	
External ID (Invoice Number)		
Delivery Date	Inbound Number	Clean All
Pur. Org.		
Plant		

Delivered (Qty)	Description (Catalog Number)	Purchase Order	SAP Number	Supplier-Item number	Material STATUS	STATUS	Cantidad esperada por la orden

Generate SAP Number

Generate Catalog Number

Figura 34: Interfaz de usuario final

A continuación, se implementó esta primera fase del programa dentro del proceso real, y trabajo perfectamente. Pero se realizó una petición extra, relacionada a añadir la opción de incluir otra base de datos para realizar la consulta.

Esta nueva base de datos, llamada "SCOPE" proviene de otra área de la empresa, y teóricamente tiene la información más actualizada en cuanto al manejo de materiales dentro de Schaeffler, por lo que se comenzó a desarrollar esta segunda opción.

Primeramente, se realizó la interfaz de usuario (se observa en la Figura 35), buscando que fuera muy similar a la mostrada anteriormente, con la misma tabla, casillas, instructivo personalizado y botones.

Instrucciones.

1. Antes de abrir este archivo, abrir SCOPE.
2. Rellenar las casillas de: Supplier, Supplier Name, External ID (Invoice Number), Delivery Date, Pur. Org., Plant.
3. Rellenar los datos de las columnas: Delivered (Qty), Material (Supplier Number), Purchase Order.
4. Si se tiene "Supplier-Item number" colocarlo en vez de Material (Catalog Number).
5. Presionar el boton "Generate SAP Number".
5. En el mensaje creado escribir el nombre completo de SCOPE (Ejemplo: SCOPE_Octubre_2023).
- 5.1 Si se tiene el SAP Number, colocarlo en la columna correspondiente "SAP Number", presionar el boton "Generate Supplier Number", y colocar el nombre del archivo.
6. Esperar a que termine el proceso.
8. Al ser "Power Automate".
9. Seleccionar el tipo "SAP - Inbound Numbers", y darle clic al boton Ejecutar.
10. Esperar a que termine el proceso.
10. Presionar el boton "Clean All" para limpiar todo.

Automation of the generation of Inbound Numbers (SCOPE)

Supplier	Supplier Name	
External ID (Invoice Number)		
Delivery Date	Inbound Number	Clean All
Pur. Org.		
Plant		

Delivered (Qty)	Material (Catalog Number)	Purchase Order	SAP Number	Supplier-Item number	Material STATUS	STATUS	Cantidad esperada por la orden

Generate SAP Number

Generate Catalog Number

Figura 35: Interfaz de Usuario SCOPE

La codificación si bien tiene similitudes, cambia bastante, esto debido a las características de este nuevo documento.

Se creo un nuevo módulo para almacenar este código, primeramente, se comenzará a trabajar con la generación de los Números de SAP, contemplando ambas posibilidades, cuando el usuario introduzca el número de catalogo o el número de proveedor.

Así que, como todo programa, se comenzaron a declarar las variables a utilizar:

```
Sub Gen_SAPNum_scope()
    Dim Libro_SCOPE As Workbook
    Dim hojaGenSAPNumber As Worksheet
    Dim hojaSCOPE As Worksheet
    Dim Catalog_Number_Column As Range
    Dim NumCatalog_Column As Range
    Dim CatalogNum_Origen As Range
    Dim CatalogNum_SCOPE As Range
    Dim nombreLibro_SCOPE As String
    Dim filaTablaSAPNum As Long

    Dim Supplier_Number_Column As Range
    Dim SupplierRef_Column As Range
    Dim Supplier_Number_Origen As Range
    Dim Supplier_Number_SCOPE As Range
```

Figura 36: Variables SCOPE

A continuación, se solicitará al usuario introducir el nombre del archivo, pero para este caso se desarrolló un procedimiento, en el que se pedirá el nombre, en caso de ser incorrecto se volverá a realizar la petición, esto se hará utilizando un bucle que se romperá hasta que se ingrese el nombre correcto, o hasta que el usuario ya no quiera ingresar el nombre.

```
' Pedir al usuario el nombre del SCOPE
Dim fileName As String
Dim userInput As Variant
Dim mostrarInputBox As Boolean
mostrarInputBox = True
Do While mostrarInputBox
    ' Mostrar el InputBox solo si mostrarInputBox es verdadero
    If mostrarInputBox Then
        userInput = InputBox("Por favor, ingrese el nombre del archivo (sin incluir la extensión):", "Nombre del archivo Excel")
    End If
    ' Verificar si el usuario ha ingresado un valor
    If userInput <> "" Then
        ' Concatenar la extensión del archivo al nombre ingresado por el usuario
        nombreLibro_SCOPE = userInput & ".xlsx"
        ' Verificar si el libro está abierto
        On Error Resume Next
        Set Libro_SCOPE = Workbooks(nombreLibro_SCOPE)
        On Error GoTo 0
        If Libro_SCOPE Is Nothing Then
            ' Mostrar el MsgBox con el botón "Introducir"
            Dim msgResponse As VbMsgBoxResult
            msgResponse = MsgBox("El libro especificado no está abierto. ¿Desea ingresar el nombre del archivo nuevamente?", vbQuestion + vbYesNo, "Abrir el SCOPE")
            ' Actualizar el valor de mostrarInputBox según la respuesta del usuario
            If msgResponse = vbNo Then Exit Sub
        Else
            ' Si el libro está abierto, salir del bucle
            mostrarInputBox = False
        End If
    Else
        ' Si el usuario no ingresa ningún valor, mostrar el MsgBox nuevamente y salir del macro
        msgResponse = MsgBox("No ha ingresado un nombre de archivo. ¿Desea ingresar el nombre del archivo nuevamente?", vbQuestion + vbYesNo, "Ingresar Nuevamente")
        If msgResponse = vbNo Then Exit Sub
    End If
Loop
```

Figura 37: Solicitar nombre SCOPE

Como se puede ver se hace uso de una variable booleana, para trabajar con un ciclo “Do While”, donde se evalúa la variable, y mientras esta sea verdadera, se repetirá indefinidamente, además se hace uso de inputbox y de msgbox para interactuar con el usuario, también se trabaja con condicionales “If”, y cuando se ha cumplido el objetivo la variable booleana se vuelve “false”, para salir del ciclo while.

En seguida, se establece la conexión con las hojas de los libros a trabajar.

```
' establecer las hojas
On Error Resume Next
Set hojaGenSAPNumber = ThisWorkbook.Sheets("Hoja_GetSAPNum_SCOPE")
Set hojaSCOPE = Libro_SCOPE.Sheets("SCOPE ")
On Error GoTo 0 ' Restaurar el manejo de errores normal
' Verificar si las hojas se establecieron correctamente
If hojaSCOPE Is Nothing Then
    ' Mostrar mensaje SI NO SE HA ENCONTRADO y salir de la macro
    MsgBox "Archivo Incorrecto, no se encuentra la hoja 'SCOPE '"
    Exit Sub
End If
```

Figura 38: Asignación de hojas SCOPE

Para seguir con el proceso, se asignan objetos a las variables declaradas anteriormente, estos objetos son columnas o rangos específicos dentro de las hojas. También se realiza una verificación, para comprobar que la tabla a trabajar tenga datos realmente, en caso contrario se utiliza el comando “Exit Sub” para salir completamente del código y terminar el programa.

```
' Definir los rangos de las columnas Material (Catalog Number) y N° Catalog en SCOPE
Set Catalog_Number_Column = hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material (Catalog Number)").DataBodyRange
Set NumCatalog_Column = hojaSCOPE.Range("C:C")

' Definir los rangos de las columnas Supplier-item number y SupplierNum en SCOPE
Set Supplier_Number_Column = hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Supplier-item number").DataBodyRange
Set SupplierRef_Column = hojaSCOPE.Range("AG:AG")

' Verificar si la tabla tiene filas de datos
If hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListRows.Count = 0 Then
    MsgBox "La tabla está vacía.", vbExclamation, "Tabla Vacía"
    Exit Sub
End If
```

Figura 39: Asignación de variables a objetos

Posteriormente se comienza a realizar una condicional, para ver que columna de datos comparar, se utiliza el “If” para verificar si la columna “Material (Catalog Number), tiene filas con datos, si es así se accede al código contenido en ese condicional, y de lo contrario se accede al código desarrollado para cuando la columna con datos es la de “Supplier-item number”.

Para la primera parte, se utilizan ciclos “for” para acceder a las filas de datos dentro de las columnas y guardarlas temporalmente en variables.

```
' Verificar si la columna Material tiene datos
If Application.WorksheetFunction.CountA(Catalog_Number_Column) > 0 Then

    ' Iterar a través de los datos en la COLUMNA Material(Catalog Number)
    For Each CatalogNum_Origen In Catalog_Number_Column
        ' Iterar a través de los datos en la columna N° Catalogo y Descripcion en SCOPE
        For Each CatalogNum SCOPE In NumCatalog_Column
```

Figura 40: Ciclo For SCOPE

A continuación, se utiliza un método nuevo, este método surge de la necesidad de acceder a una cadena de caracteres, esto es por qué en la base de datos conocida como “SCOPE”, en la columna donde se encuentran los números de catálogo, los números vienen acompañados de una pequeña descripción, y esto para fines de comparación impide que se encuentren coincidencias, los números tienen el formato visible en la Figura 41.

N°Catálogo y Descripción			
630		00	-Kit embrague
624		00	-Kit embrague
542		00	-Bomba direc.

Figura 41: Numero de Catalogo SCOPE

Por lo que en el código se comienza a buscar el “-“ para poder dividir la cadena y almacenar únicamente el número.

```
If InStr(1, CatalogNum_SCOPE.Value, "-") > 0 Then ' Verif
    ' Obtener la parte antes del guion ("-") en CatalogNum_SCOPE
    Dim parteAntesDelGuion As String
    parteAntesDelGuion = Trim(Split(CatalogNum_SCOPE.Value, "-")(0))
```

Figura 42: Dividir Cadena

El comando InStr devuelve un valor Variant (Long) que especifica la posición de la primera ocurrencia de una cadena dentro de otra. Lo que se busca es saber si existe el “-“ para realizar la comparación, a continuación, se declara una variable para almacenar únicamente el número, el comando “Split” se utiliza para dividir la cadena entre el número y lo consecuente al guion, y la función “Trim” arroja el numero sin incluir los espacios que siguen a él.

En seguida, una vez que ya se puede comparar el número, se realiza la comparación, cuando se encuentra la coincidencia, se comienzan a asignar los valores determinados para cada una de las columnas. En la figura 43 se observa dicha codificación.

```
Iterar a través de los datos en la COLUMNA Material(Catalog Number)
For Each CatalogNum_Origen In Catalog_Number_Column
    ' Iterar a través de los datos en la columna N° Catalogo y Descripcion en SCOPE
    For Each CatalogNum_SCOPE In NumCatalog_Column
        ' Verificar si la CatalogNum_SCOPE contiene un guion ("-")
        If InStr(1, CatalogNum_SCOPE.Value, "-") > 0 Then ' Verificar si el valor de la celda CatalogNum_SCOPE contiene el carácter "-"
            ' Obtener la parte antes del guion ("-") en CatalogNum_SCOPE
            Dim parteAntesDelGuion As String
            parteAntesDelGuion = Trim(Split(CatalogNum_SCOPE.Value, "-")(0)) ' divide la cadena CatalogNum_SCOPE.Value en partes, usando el guion "-" como delimitador. Y TRIM elimina espacio
            ' Comprobar si la parte antes del guion de la CatalogNum_SCOPE coincide con CatalogNum_Origen
            If UCase(parteAntesDelGuion) = UCase(CatalogNum_Origen.Value) Then ' Comparación sin distinguir mayúsculas y minúsculas
                ' Obtener el índice de fila en la Tabla_NumerosSAP
                filaTablaSAPNum = CatalogNum_Origen.Row - Catalog_Number_Column.Row + 1
                ' Copiar datos a la COLUMNA SAP NUMBER
                hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("SAP Number").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = hojaSCOPE.Cells(CatalogNum_SCOPE.Row, 2).Value
                ' Copiar datos a la COLUMNA Supplier - item number
                hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Supplier-item number").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = "" & hojaSCOPE.Cells(CatalogNum_SCOPE.Row, 33)
                ' Copiar datos a la COLUMNA Material STATUS
                hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material STATUS").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = hojaSCOPE.Cells(CatalogNum_SCOPE.Row, 9).Value
                ' Cambiar el color de fondo de la celda si Material STATUS es diferente de 61
                If hojaSCOPE.Cells(CatalogNum_SCOPE.Row, 9).Value <> 61 Then
                    hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material STATUS").DataBodyRange.Cells(filaTablaSAPNum, 1).Interior.Color = RGB(255, 0, 0) ' Rojo
                End If
            End If
        Exit For ' Salir del bucle si se encontró una coincidencia
    End If
End If
Next CatalogNum_SCOPE
Next CatalogNum_Origen
```

Figura 43: Asignación de valores SCOPE

Como se pudo observar en la Figura anterior, en este caso se decidió obtener el índice de la fila para poder trabajar con la tabla del “SCOPE”, esto debido a que en esta base de datos no hay una tabla declarada como tal. Para realizar la asignación de los valores, se fue accediendo a las posiciones utilizando las variables de las columnas, y accediendo a las mismas mediante su nombre, y colocando su valor mediante el comando “Cells”, que accede a una celda específica, en este caso la celda se encontraba en la columna indicada en el código y la fila es la variable creada para acceder a las filas de la tabla.

Además, se incluyó también la funcionalidad de colorear el estatus de los materiales que fueran distintos al ya mencionado “61”.

A continuación, se realizó la codificación adecuada cuando la columna con datos fuera la “Supplier-item number”, es prácticamente igual a la mostrada anteriormente, solo que no incluye la comparación de cadenas de texto con guiones incluidos.

```
' Iterar a través de los datos en la COLUMNA supplier - item number
For Each Supplier_Number_Origen In Supplier_Number_Column
    ' Iterar a través de los datos en la columna supplier reference en SCOPE
    For Each Supplier_Number_SCOPE In SupplierRef_Column
        ' Verificar si la Supplier_Number_SCOPE contiene un guion ("-")
        If CStr(Supplier_Number_SCOPE.Value) = CStr(Supplier_Number_Origen.Value) Then ' Comparación sin distinguir mayúsculas y minúsculas
            ' Obtener el índice de fila en la Tabla_NumerosSAP
            filaTablaSAPNum = Supplier_Number_Origen.Row - Supplier_Number_Column.Row + 1
            ' Copiar datos a la COLUMNA SAP NUMBER
            hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("SAP Number").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = hojaSCOPE.Cells(Supplier_Number_SCOPE.Row, 2).Value

            If InStr(1, hojaSCOPE.Cells(Supplier_Number_SCOPE.Row, 3).Value, "-") > 0 Then ' Verificar si el valor de la celda CatalogNum_SCOPE contiene el carácter "-"
                ' Obtener la parte antes del guion ("-") en CatalogNum_SCOPE
                Dim parteAntesDelGuion2 As String
                parteAntesDelGuion2 = Trim(Split(hojaSCOPE.Cells(Supplier_Number_SCOPE.Row, 3).Value, "-")(0))
                ' Copiar datos a la COLUMNA Supplier - item number
                hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material (Catalog Number)").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = parteAntesDelGuion2
                End If

                ' Copiar datos a la COLUMNA Material_STATUS
                hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material_STATUS").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = hojaSCOPE.Cells(Supplier_Number_SCOPE.Row, 9).Value
                ' Cambiar el color de fondo de la celda si Material_STATUS es diferente de 61
                If hojaSCOPE.Cells(Supplier_Number_SCOPE.Row, 9).Value <> 61 Then
                    hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material_STATUS").DataBodyRange.Cells(filaTablaSAPNum, 1).Interior.Color = RGB(255, 0, 0) ' Rojo
                End If
            Exit For ' Salir del bucle si se encontró una coincidencia
        End If
    Next Supplier_Number_SCOPE
Next Supplier_Number_Origen
```

Figura 44: Asignación de valores SCOPE_supplier

Como se puede observar en la Figura 44, la única variación ocurre al momento de colocar el número de catálogo, donde se utiliza un método similar para dividir la cadena de caracteres y colocar únicamente el número.

Finalmente se añadió un MsgBox que permite mostrar al usuario cuando el proceso se ha completado.

```
End If

' Mostrar mensaje de éxito
MsgBox "Se ha realizado correctamente.", vbInformation, "Proceso Terminado"

End Sub
```

Figura 45:Mensaje de éxito SCOPE

Del mismo modo se desarrolló la codificación que permite obtener el número de catalogo y de proveedor cuando se introduce el número de SAP, este código es prácticamente igual, con la diferencia de que se cambian los objetos, ya sean columnas o rango de las hojas de trabajo. En la figura 46 se observa que primero se declaran variables, y se realiza la misma petición del nombre del archivo.

```
Sub Gen_SupplierNumber_scope()
Dim Libro_SCOPE As Workbook
Dim hojaGenSAPNumber As Worksheet
Dim hojaSCOPE As Worksheet
Dim SAP_Number_Column As Range
Dim Num_SAP_SCOPE_Column As Range
Dim SAPNum As Range
Dim NumSAP_SCOPE As Range
Dim nombreLibro_SCOPE As String
Dim filaTablaSAPNum As Long

' Pedir al usuario el nombre del SCOPE
Dim fileName As String
Dim userInput As Variant
Dim mostrarInputBox As Boolean
mostrarInputBox = True
Do While mostrarInputBox
    ' Mostrar el InputBox solo si mostrarInputBox es verdadero
    If mostrarInputBox Then
        userInput = InputBox("Por favor, ingrese el nombre del archivo (sin incluir la extensión):", "Nombre del archivo Excel")
    End If
    ' Verificar si el usuario ha ingresado un valor
    If userInput <> "" Then
        ' Concatenar la extensión del archivo al nombre ingresado por el usuario
        nombreLibro_SCOPE = userInput & ".xlsx"
        ' Verificar si el libro está abierto
        On Error Resume Next
        Set Libro_SCOPE = Workbooks(nombreLibro_SCOPE)
        On Error GoTo 0
        If Libro_SCOPE Is Nothing Then
            ' Mostrar el MsgBox con el botón "Introducir"
            Dim msgResponse As VbMsgBoxResult
            msgResponse = MsgBox("El libro especificado no está abierto. ¿Desea ingresar el nombre del archivo nuevamente?", vbQuestion + vbYesNo, "Abrir el SCOPE")
            ' Actualizar el valor de mostrarInputBox según la respuesta del usuario
            If msgResponse = vbNo Then Exit Sub
        Else
            ' Si el libro está abierto, salir del bucle
            mostrarInputBox = False
        End If
    Else
        ' Si el usuario no ingresa ningún valor, mostrar el MsgBox nuevamente y salir del macro
        msgResponse = MsgBox("No ha ingresado un nombre de archivo. ¿Desea ingresar el nombre del archivo nuevamente?", vbQuestion + vbYesNo, "Ingresar Nuevamente")
        If msgResponse = vbNo Then Exit Sub
    End If
Loop
```

Figura 46: Variables y Nombre de libro SCOPE

A continuación, se asignan los objetos a las variables, y se realiza la comprobación para verificar si la tabla tiene datos, como se observa en la Figura 47.

```
' Intentar establecer las hojas
On Error Resume Next
Set hojaGenSAPNumber = ThisWorkbook.Sheets("Hoja_GetSAPNum_SCOPE")
Set hojaSCOPE = Libro_SCOPE.Sheets("SCOPE ")
On Error GoTo 0 ' Restaurar el manejo de errores normal
' Verificar si las hojas se establecieron correctamente
If hojaSCOPE Is Nothing Then
    ' Mostrar mensaje y salir de la macro
    MsgBox "Archivo Incorrecto, no se encuentra la hoja 'SCOPE '"
    Exit Sub
End If

' Definir los rangos de las columnas SAP Number (Origen) y N°SAP en SCOPE
Set SAP_Number_Column = hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("SAP Number").DataBodyRange
Set Num_SAP_SCOPE_Column = hojaSCOPE.Range("B:B")

' Verificar si la tabla tiene filas de datos
If hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListRows.Count = 0 Then
    MsgBox "La tabla 'Tabla_NumerosSAP' está vacía.", vbExclamation, "Tabla Vacía"
    Exit Sub
End If

' Verificar si la columna SAP Number tiene filas de datos
If WorksheetFunction.CountA(hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("SAP Number").DataBodyRange) = 0 Then
    MsgBox "La columna SAP Number está vacía.", vbExclamation, "Columna sin datos"
    Exit Sub
End If
```

Figura 47: Establecer Objetos SCOPE

Y finalmente se realiza la comparativa y asignación de datos, basándose en los ciclos “for” para recorrer las columnas y comparar los valores, para seguidamente asignar los valores correspondientes en todas las columnas indicadas.

```
' Iterar a través de los datos en la hoja GenSAPNumber
For Each SAPNum In SAP_Number_Column
' Iterar a través de los datos en la hoja SCOPE
For Each NumSAP_SCOPE In Num_SAP_SCOPE_Column
' Comprobar si la cadena de la NumSAP_SCOPE contiene la cadena de SAPNum
If StrComp(NumSAP_SCOPE.Value, SAPNum.Value, vbTextCompare) = 0 Then
' Obtener el índice de fila en la Tabla NumerosSAP
filaTablaSAPNum = SAPNum.Row - SAP_Number_Column.Row + 1
' Obtener el valor antes del guion ("-") en la columna N°SAP de la hoja SCOPE
Dim valor As String
valor = Trim(Split(NumSAP_SCOPE.Offset(0, 1).Value, "-")(0))
' Copiar datos a la columna Material (Catalog Number)
hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material (Catalog Number)").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = valor

' Copiar datos a la COLUMNA Supplier - item number
hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Supplier-item number").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = hojaSCOPE.Cells(NumSAP_SCOPE.Row, 33).Value

' Copiar datos a la columna Material STATUS
hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material STATUS").DataBodyRange.Cells(filaTablaSAPNum, 1).Value = hojaSCOPE.Cells(NumSAP_SCOPE.Row, 9).Value
' Cambiar el color de fondo de la celda si Material STATUS es diferente de 61
If hojaSCOPE.Cells(NumSAP_SCOPE.Row, 9).Value <> 61 Then
hojaGenSAPNumber.ListObjects("Tabla_NumerosSAP").ListColumns("Material STATUS").DataBodyRange.Cells(filaTablaSAPNum, 1).Interior.Color = RGB(255, 0, 0) ' Rojo
End If
Exit For ' Salir del bucle si se encontró una coincidencia
End If
Next NumSAP_SCOPE
Next SAPNum
' Mostrar mensaje de éxito
MsgBox "Se ha realizado correctamente.", vbInformation, "Proceso Terminado"
End Sub
```

Figura 48: Asignación de Valores SCOPE con SAP Numbers

Al igual que con la otra base de datos, en esta interfaz también se incluyó un botón que permite limpiar la tabla, y se implementó con el código visible en la Figura 49:

```
Sub Clean_scope()
' Borrar datos de las casillas específicas
ThisWorkbook.Sheets("Hoja_GetSAPNum_SCOPE").Range("L2,L3,L4,L5,L6,N2,N4").ClearContents

' Borrar filas, datos y formato de la tabla "Tabla_SAPNum"
Dim wsGenSAPNumber As Worksheet
Dim tablaSAPNum As ListObject

' Referenciar la hoja y la tabla
Set wsGenSAPNumber = ThisWorkbook.Sheets("Hoja_GetSAPNum_SCOPE")
Set tablaSAPNum = wsGenSAPNumber.ListObjects("Tabla_NumerosSAP")

' Borrar filas y formato de la tabla
If Not tablaSAPNum.DataBodyRange Is Nothing Then

tablaSAPNum.DataBodyRange.Clear
tablaSAPNum.DataBodyRange.Delete

End If
End Sub
```

Figura 49: Limpiar Tabla SCOPE

Como se puede visualizar es muy similar la codificación a la utilizada en la otra interfaz, donde se asignan las hojas y las tablas, y seguidamente se verifica que no estén vacías, para proceder a limpiarlas y a eliminar las filas a continuación.

Para tener una interfaz más agradable para el usuario se realizó un formulario de usuario, que permitiera elegir que fuente de información o base de datos le gustaría utilizar, se decidió utilizar optionButtons para limitar las respuestas del usuario, y se decidió utilizar un botón para confirmar una selección. La interfaz quedó como se ve en la Figura 50:



Figura 50: UserForm

El código utilizado para relacionar las hojas con el botón fue el mostrado en la Figura 51. Primero se añadió la codificación necesaria para mostrar el formulario al abrir el archivo.

```
Private Sub Workbook_Open()
    ' Mostrar el formulario
    UserForm_Inicio.Show vbModal

    ' Restaurar el valor de botonPresionado para futuros usos
    botonPresionado = False
End Sub
```

Figura 51: Abrir el libro

Donde se accede al Sub que se activa al abrir un libro en Excel, y en seguida se utiliza el método “Show”, para mostrar el formulario indicado.

La codificación utilizada para acceder a las hojas se basa en la acción del botón “OK”, donde se relacionan los radioButtons con las hojas específicas para cada interfaz.

```
Public botonPresionado As Boolean

Private Sub CommandButton1_Click()
    ' Obtiene el valor del radio button seleccionado
    Dim radioButtonValue As String
    If OptionButton1.Value = True Then
        radioButtonValue = "OptionButton1"
    Else
        radioButtonValue = "OptionButton2"
    End If

    ' Muestra la hoja correspondiente al radio button seleccionado
    If radioButtonValue = "OptionButton1" Then
        Sheets("Hoja_GetSAPNum_SCOPE").Activate
    Else
        Sheets("Hoja_GenSAPNumber_MMav").Activate
    End If

    ' Establece la variable botonPresionado en True
    botonPresionado = True

    ' Oculta el formulario después de realizar las operaciones
    Me.Hide
End Sub
```

Figura 52: Código UserForm

Y adicionalmente se decidió añadir un método para que el usuario seleccione alguna opción.

```
Private Sub UserForm_QueryClose(Cancel As Integer, CloseMode As Integer)
    ' Verifica si el botón ha sido presionado
    If Not botonPresionado Then
        ' Si el botón no ha sido presionado, cancela el cierre del formulario
        Cancel = True
        MsgBox "Por favor, seleccione una opcion.", vbExclamation
    End If
End Sub
```

Figura 53: Seleccionar Base de datos

Como se puede observar se decidió incluir una condicional “If”, donde se verifica si el usuario presiona el botón “Ok” del formulario, y de lo contrario se hace que el formulario no se cierre hasta que una opción haya sido seleccionada, arrojando un mensaje al usuario indicando lo que debe realizar.

A continuación, se comenzó a desarrollar la integración con el sistema de SAP de Schaeffler, para esto se tuvo que solicitar una gran cantidad de permisos dentro de la empresa.

Se desarrollo esta conexión utilizando el software de Power Automate, debido a que es el software en el que Schaeffler permite a sus trabajadores realizar automatizaciones.

Primeramente, se creó un nuevo flujo de escritorio, para permitir automatizar tareas dentro del propio escritorio.

Ya dentro del flujo se comenzó a desarrollar la lógica de programación, por supuesto, al utilizar Power Automate se utiliza programación “Low Code”, por lo que es necesario utilizar los flujos preestablecidos para realizar toda esta programación.

Se utilizó el siguiente comando, para establecer una conexión con el archivo de “Excel” en donde se encuentra toda la información con la que se va a trabajar, se decidió utilizar este comando, ya que permite la conexión con un libro de Excel que se encuentre abierto dentro del computador.



Figura 54: Iniciar Excel PA

Se consideró el incluir un Cuadro de mensaje, que diera la opción al trabajador de seleccionar una opción para seleccionar la hoja de trabajo donde se encuentra almacenada la información, esto de acuerdo con las bases de datos utilizadas en el primer parte del proceso.

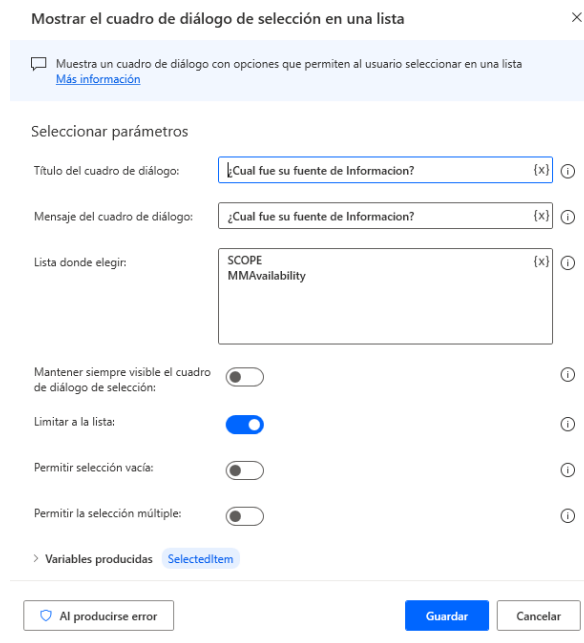


Figura 55: Cuadro de mensaje PA

Se decidió incluir un condicional, que utilizara la variable generada con el cuadro de mensaje anterior, para asignar la hoja de trabajo activa. El comando “If”, que se comporta exactamente igual que cualquier condicional de algún lenguaje de programación, compara la opción seleccionada por el usuario, por una de las opciones, y cuando se cumple la condicional, utiliza el flujo llamado “Establecer la hoja de cálculo de Excel activa”, y como se observa en la Figura mostrada a continuación se asigna el nombre de la hoja para ambas condiciones.

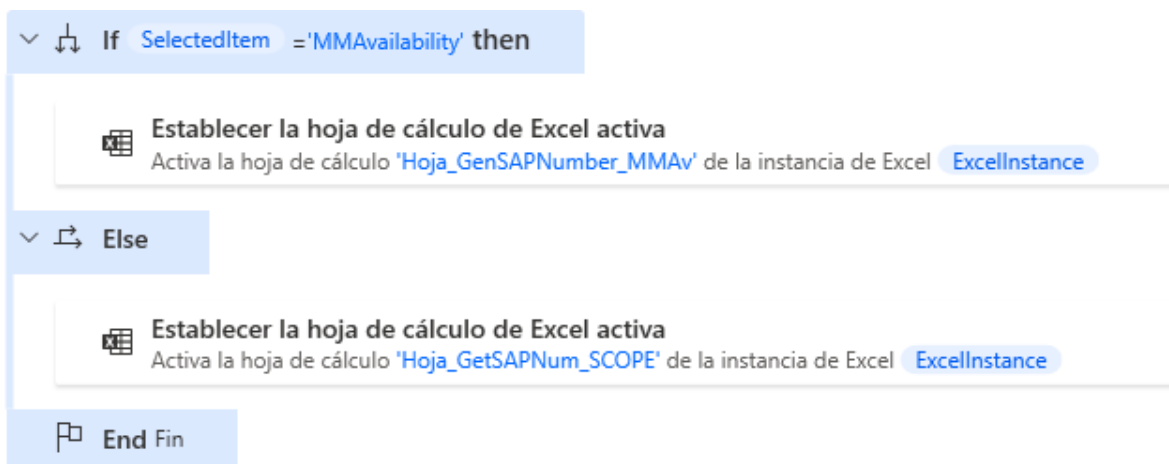


Figura 56: Hoja de Trabajo Activa PA

En seguida es necesario comenzar a almacenar los datos que deben ser ingresados en la transacción en variables, para ello se utiliza el comando de Power Automate, llamado: “Leer en hoja de cálculo de Excel”, primeramente es necesario leer y almacenar los datos que pueden ser considerados una constante en el proceso dentro de la transacción, estos datos son: “Supplier”, “External ID”, “Delivery Date”, “Purchase Organization” y “Plant”, como se observa en la Figura mostrada a continuación se utiliza la columna y el número de fila de cada dato específico, y después se crean variables y se utilizan para almacenar estos datos.

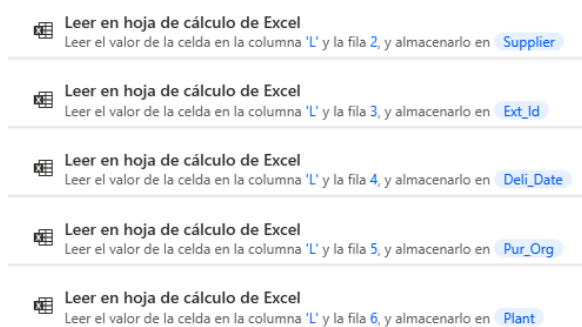


Figura 57: Variables Constantes PA

Para continuar, es necesario comenzar a trabajar con los datos almacenados en la tabla, es decir, con los números de Material, las cantidades y las órdenes de compra a las que se

referencia ese material. Por lo que es necesario buscar un comando que sea capaz de buscar la última fila con datos dentro de la tabla, para poder trabajar con la tabla como tal, esto debido a que Power Automate, no tiene incorporado dentro de sus comandos, algún comando que sea capaz de identificar las tablas por su nombre, y mucho menos de identificar el número de filas o columnas de las tablas en general.

Se decide utilizar el comando llamado “Obtener la primera fila libre de la columna de la hoja de cálculo de Excel”, y se le asigna la columna en la que se encuentran los números de Material, sabiendo que la fila que se encuentre será una unidad más grande que el número de columnas que maneja la tabla.

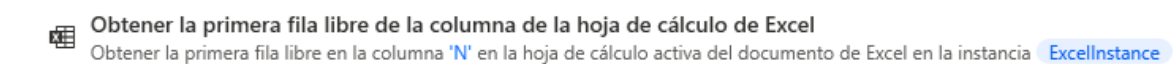


Figura 58: Obtener última fila PA

A continuación, es necesario almacenar la información de la tabla, por lo que se crearan 3 variables que permitan esto, y se utilizara el comando llamado “Leer en hoja de cálculo de Excel”, y se seleccionara la opción que permite leer un rango de celdas, y el rango estará delimitado por el comienzo de la tabla, que es un valor que conocemos, y el final de la tabla, que es el valor que calculamos anteriormente, restándole una unidad.

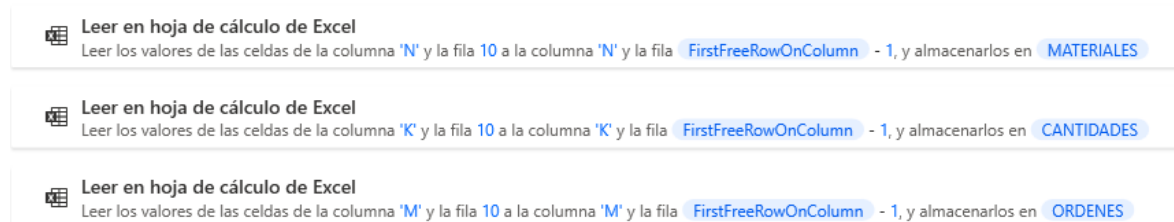


Figura 59: Almacenar columnas PA

Como se puede ver en la Figura 59, es necesario repetir este comando 3 veces distintas para poder obtener los valores de la tabla, “Materiales”, “Cantidades” y “Órdenes de Compra”.

En la Figura 60, se observa que se utiliza un comando que permite acceder a la transacción en SAP directamente, y se utiliza el comando llamado “Esperar la ventana” para permitir que la transacción se abra correctamente y permitir que el flujo pueda continuar ejecutándose de una manera adecuada.

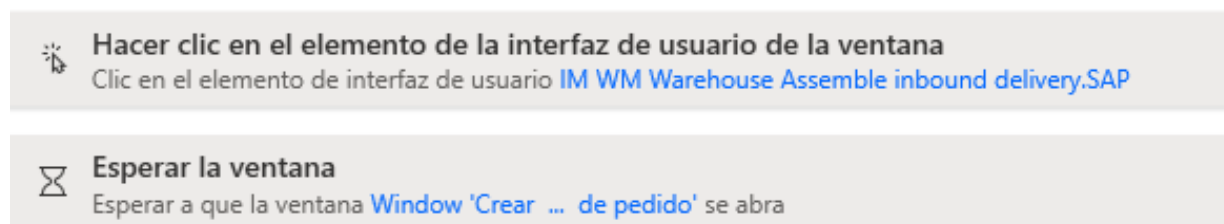


Figura 60: Abrir Transacción

Y en seguida se comienza a realizar la carga de información, para esto fue necesario utilizar una herramienta de Power Automate, llamada “Grabadora”, que permite al usuario grabar distintas acciones dentro del escritorio y escribirlas todas en forma de comandos, esta herramienta fue utilizada debido a que es una manera practica de obtener el nombre de cada elemento dentro de la transacción, y como se observa en la Figura 61, mostrada a continuación, cada elemento tiene un identificador diferente.



Figura 61: Rellenar campos en la transacción

Es posible observar en la Figura 61, que después de obtener cada elemento dentro de la transacción, se utiliza el comando “Rellenar el campo de texto en la ventana”, para escribir en cada elemento de la transacción las variables que se almacenaron inicialmente como constantes en el proceso.

Ahora comienza la parte más complicada, donde es necesario realizar la carga de todos los materiales almacenados en la tabla, por supuesto con su cantidad especifica, y verificando siempre que coincida con la orden de compra referenciada, para hacer esto se decidió utilizar un bucle, originalmente se había estado tratando de utilizar un bucle llamado “For Each” anidado, donde se accediera a los datos almacenados en ambos conjuntos de datos (Materiales y Cantidades), y dentro del bucle se realizara la comparativa con la orden de compra, pero este método se descartó, ya que dentro de Power Automate, este bucle tiene un comportamiento algo distinto a lo mostrado en Visual Basic, y al ejecutar en esta plataforma este bucle, lo único que se obtenía es que se realizaran las iteraciones primeramente de un conjunto de datos, y en seguida del otro.

Finalmente se decidió utilizar el comando llamado “Bucle” dentro de la plataforma, este bucle tiene una estructura muy similar al bucle tradicional llamado “For”, en donde es necesario asignar parámetros, como el numero o la variable en la cual iniciar el conteo, el numero o variable en la cual terminarlo, y el valor a tomar para ir aumentando, en este caso el conteo debe comenzar en 0, y debe terminar en el número de filas de alguna de la columna almacenada anteriormente, menos 1, para terminar en el número de filas de la tabla, por lo que se utiliza el siguiente comando “`%MATERIALES.RowsCount - 1%`”

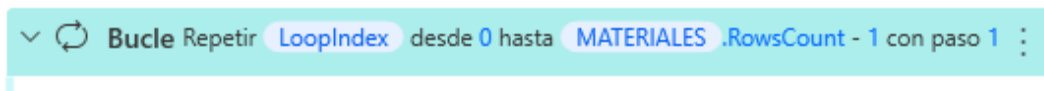


Figura 62: Bucle PA

Dentro del bucle es necesario incluir la instrucción para ir cargando todos los materiales y sus respectivas cantidades, para ello se utiliza el comando mostrado en la Figura 63.



Figura 63: Rellenar Materiales PA

Como se puede ver en la Figura 63, se utilizó el mismo comando que se usó en los valores constantes, con la única diferencia que en este caso se asignó la variable del ciclo llamada “LoopIndex”, para ir accediendo una por una a la fila de los materiales y a las cantidades dentro de la tabla.

A continuación, se presiona la tecla “Enter”, con ayuda del comando mostrado en la siguiente Figura, y enseguida se utiliza un flujo llamado “Espera”, que permite frenar el proceso, y permitir que los datos dentro de la transacción se carguen adecuadamente.

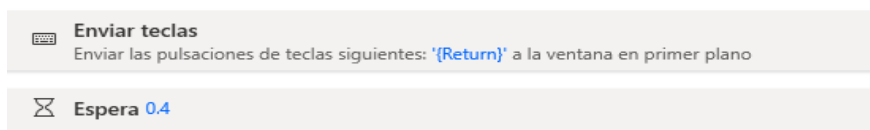


Figura 64: Enviar Tecla PA

Para continuar con el proceso es necesario almacenar los valores que se muestran como las órdenes de compra dentro de la transacción, para ello se utiliza el comando llamado “Obtener detalles del elemento de interfaz de usuario en la ventana”, como se muestra en la Figura 65, se accede a cada elemento mediante su identificador, y después se almacena su valor dentro de variables creadas.



Figura 65: Almacenar Órdenes de Compra PA

Teniendo los valores disponibles como órdenes de compra, ya es posible realizar la comparativa para seleccionar alguna opción, para ello se hará uso del comando condicional llamado “If”. Para hacer la comparativa se accede a la fila con la variable “LoopIndex”, del

conjunto de datos de las órdenes de compra, y se compara con cada uno de los elementos dentro de la interfaz de usuario que muestran esta información.

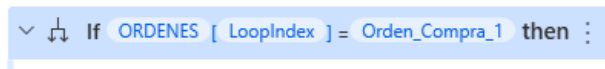


Figura 66: Condicional Órdenes de Compra

Cuando se encuentra la coincidencia, es necesario modificar el valor de otra casilla dentro de la transacción, para ello se utiliza el comando mostrado en la Figura 67.

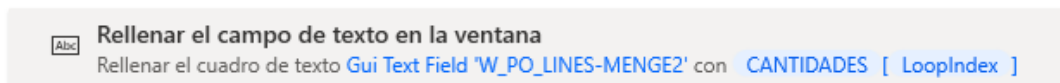


Figura 67: Modificar Cantidad PA

Y es necesario hacer clic en otra casilla para seleccionar la opción correcta, por ello se utiliza el comando mostrado a continuación, donde se observa que se accede a dicha casilla por su identificador.



Figura 68: Clic de selección PA

En la Figura 69, mostrada a continuación se visualiza el flujo utilizado para presionar el botón “Adopt Items” dentro de la transacción.

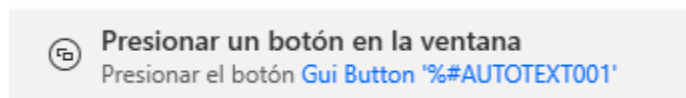


Figura 69: botón Adopt Items PA

Además, para permitir al trabajador monitorear la forma en que se realizó el proceso, y las cantidades de espera, como de llegada, es necesario almacenar el valor mostrado en una casilla dentro de la transacción, que permite ver la cantidad esperada, y enseguida se utiliza el flujo “Escribir en la hoja de cálculo de Excel”, para escribir en la columna “STATUS” la cantidad de entrada y la cantidad esperada.

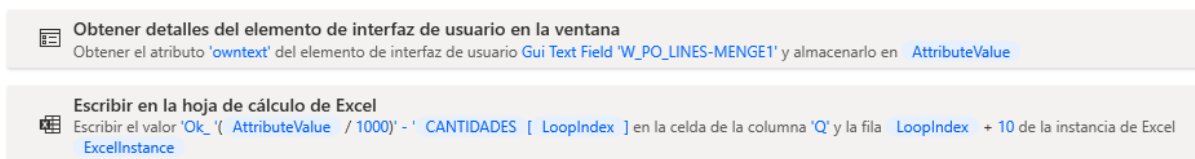


Figura 70: Escribir Cantidades

Como se visualiza en la Figura 70 se almacena el valor de la cantidad mostrada en la transacción en una variable, y en el siguiente flujo se utiliza esa misma variable, para ser

mostrada con la cantidad de entrada, y el texto “Ok”, que indica que se realizó de forma correcta el proceso.

Este proceso con la condicional es repetido utilizando el flujo “Else If”, para todas las órdenes de compra mostradas en la transacción.



Figura 71: Condicionales Órdenes de Compra PA

Finalmente, si no se encuentra coincidencia con alguna orden de compra, se utiliza el flujo “Else”, para abarcar todas las condiciones que no se han mencionado, y dentro de este flujo es necesario presionar la tecla “Enter”, para que la transacción permita ingresar algún otro número de material, y repetir el proceso las veces que sean necesarias. Y se escribe en la columna “STATUS” del libro donde se ha estado trabajando el texto “Sin PO”, para indicar que ese número de material no cuenta con una orden de compra abierta. En la Figura 72 se muestra esto:

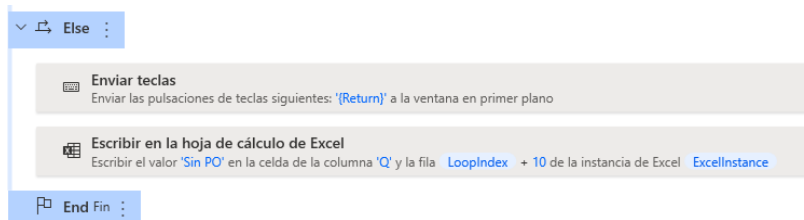


Figura 72: Condicional Else PA

Y a continuación, este proceso es repetido para todos los materiales contenidos en la tabla, y el bucle finaliza con el comando mostrado en la Figura 73.



Figura 73: Final del bucle

CAPÍTULO IV

RESULTADOS

Al concluir con el desarrollo de la aplicación se obtuvo el formulario mostrado en la Figura 74, este mismo es mostrado al iniciar, para permitirle al usuario seleccionar la fuente de información con la que desea trabajar, donde se observa la posibilidad de elegir entre dos opciones.

Fuente de Informacion ×

Seleccione la fuente de Informacion de donde desea obtener los datos

SCOPE

MM

Availability

OK

Figura 74: Formulario Selección BD

Se desarrollaron dos interfaces distintas para permitir al usuario trabajar con ambas fuentes de información, la primera es la mostrada en la Figura 75.

Instrucciones.

1. Antes de abrir este archivo, abrir SCOPE.
2. Rellenar las casillas de: Supplier, Supplier Name, External ID (Invoice Number), Delivery Date, Pur. Org., Plant.
3. Rellenar los datos de las columnas: Delivered (Qty), Material (Supplier Number), Purchase Order.
4. O si se tiene "Supplier-Item number" colocarlo en vez de Material (Catalog Number).
5. Presionar el boton "Generate SAP Number"
6. En el mensaje creado escribir el nombre completo de SCOPE. (Ejemplo: SCOPE, Octubre 2023)
7. Si se tiene el SAP Number, colocarlo en la columna correspondiente "SAP Number", presionar el boton "Generate Supplier Number", y colocar el nombre del archivo.
8. Esperar a que termine el proceso.
9. Presionar el boton "Clean All" para limpiar todo.

Automation of the generation of Inbound Numbers (SCOPE)

Supplier		Supplier Name				
External ID (Invoice Number)		Inbound Number				
Delivery Date						
Pur. Org.						
Plant						

Clean All

Delivered (Qty)

Material (Catalog Number)

Purchase Order

SAP Number

Supplier-Item number

Material STATUS

STATUS

Generate SAP Number

Generate Catalog Number

Figura 75: Interfaz de Usuario SCOPE

A continuación, se cargarán los números de Catalogo de los materiales en la columna adecuada, además de las cantidades estipuladas y su correspondiente orden de compra. Por lo que ya cargada la información se verá como se muestra en la Figura 78.

Instrucciones.

1. Antes de abrir este archivo, abrir MM Availability.
2. Rellenar las casillas de: External ID (Invoice Number), Delivery Date, Pur. Org., Plant.
3. Rellenar los datos de las columnas: Delivered (Qty), Description (Catalog Number) y Purchase Order (0 si se tiene Supplier-item number en vez de Description (Catalog Number))
4. Copiar el nombre completo de MM Availability, (Ejemplo: MM Availability_0145_CW40_2023)
5. Presionar el botón "Generate SAP Number"
6. En el mensaje creado pegar el nombre del archivo antes copiado.
- 6.1 Si se tiene el SAP Number, colocarlo en la columna correspondiente "SAP Number", presionar el botón "Generate Catalog Number", y colocar el nombre del archivo.
7. Esperar a que termine el proceso.
8. Abrir Power Automate
9. Seleccionar el flujo "SAP - Inbound Numbers", y darle clic al botón Ejecutar.
10. Esperar a que termine el proceso.
11. Presionar el botón "Clean All" para limpiar todo.

Automation of the generation of Inbound Numbers (MM Availability)							
Supplier		Supplier Name					
External ID (Invoice Number)	AB151						
Delivery Date	02.02.24	Inbound Number					
Pur. Org.	6145						
Plant	0145						
		Clean All					
Delivered (Qty)	Description (Catalog Number)	Purchase Order	SAP Number	Supplier-item number	Material-STATUS	STATUS	Cantidad esperada por la orden
131	D85229	00054					
204	D88795	00054					
248	D10782	00054					
189	CMD256	00054					
180	D85275	00054					
160	D47788	00054					
160	CMD280	00054					
150	CMD368	00054					
140	D14882	00054					
120	D15486	00054					
120	D16989	00054					
94	CMD194	00054					
85	CMD135	00054					
82	CMD184	00054					
63	CMD172	00054					
40	D15383	00054					
165	D17444	00054					
55	D16787	00054					
54	CMD193	00054					
49	CMD201	00054					
49	CMD165	00054					
49	D17478	00054					
42	D14773	00054					
50	D15563	00054					
3	D15588	00054					
40	D17783	00054					
49	CMD433	00054					
160	CMD164	00054					
140	CMD180	00054					
180	D15931	00054					
90	CMD177	00054					
70	D16289	00054					
160	D16363	00054					
40	D10275	00054					
44	D12583	00054					
39	D17883	00054					
10	D25772	00054					
49	D10173	00054					

Generate SAP Number

Generate Catalog Number

Figura 78: Plantilla con información cargada

A continuación, se realizará el proceso para la obtención de los números de SAP para cada número de material, además de obtener el estatus del material, y poder verificar que no existan problemas antes de realizar la carga en la transacción de SAP. Para ello se presionará el botón "Genérate SAP Numbers", y en el inputbox desplegado, se colocará el nombre del archivo que funciona como base de datos. En la Figura 79 se observa lo antes mencionado.

Nombre del archivo Excel

Por favor, ingrese el nombre del archivo (sin incluir la extensión):

MM Availability_0145_CW01_2024

Aceptar

Cancelar

Generate SAP Number

Generate Catalog Number

Figura 79: Introducir Base de Datos

Después se presiona el botón de “Aceptar” y se espera a que termine el proceso. Y se puede observar en la siguiente Figura que ya se ha cargado la información del proveedor, es decir, su número de identificación, y se han cargado también los números de “SAP” y los números de proveedor, además del estatus actual del material.

Instrucciones.

1. Antes de abrir este archivo, abrir MM Availability.
2. Rellenar las casillas de: External ID (Invoice Number), Delivery Date, Pur. Org., Plant.
3. Rellenar los datos de las columnas: Delivered (Qty), Description (Catalog Number) y Purchase Order (O si se tiene Supplier-item number en vez de Description (Catalog Number)).
4. Copiar el nombre completo de MM Availability (Ejemplo: MM Availability 0145 CW40 2023).
5. Presionar el botón "Generate SAP Number".
6. En el mensaje creado pegar el nombre del archivo antes copiado.
- 6.1. Si se tiene el SAP Number, colocarlo en la columna correspondiente "SAP Number", presionar el botón "Generate Catalog Number", y colocar el nombre del archivo.
7. Esperar a que termine el proceso.
8. Abrir Power Automate.
9. Seleccionar el flujo "SAP - Inbound Numbers", y darle clic al botón Ejecutar.
10. Esperar a que termine el proceso.
11. Presionar el botón "Clean All" para limpiar todo.

Automation of the generation of Inbound Numbers (MM Availability)

Supplier: 260 Supplier Name: Auto Manufacturas Drake S.A. de C.V.

External ID (Invoice Number): ABT Inbound Number: [Empty]

Delivery Date: 02/02/24 Pur. Org.: 6145 Plant: 0045

Clean All

Delivered (Qty)	Description (Catalog Number)	Purchase Order	SAP Number	Supplier-item number	Material STATUS	STATUS	Cantidad esperada por la orden
128	D15359	000558	0899250	8804 C	E1		
204	D68979	000558	0564235	7563 C	E1		
249	D18759	000558	0764702	8232 C	E1		
388	CMD091	000558	0880737	7389 C	E1		
388	D68279	000558	0564219	7532 C	E1		
90	D17199	000558	0674670	8271 C	E1		
90	CMD289	000558	0880747	7384 C	E1		
90	CMD341	000558	0880739	7563 C	E1		
140	D18492	000558	0779453	8298 C	E1		
128	D14548	000558	0779399	8653 C	E1		
120	D18999	000558	0853953	8994 C	E1		
34	CMD109	000558	0949250	9125 C	E1		
85	CMD103	000558	0828045	8502 C	E1		
63	CMD101	000558	0828143	8076 C	E1		
63	CMD107	000558	0828919	8852 C	E1		
40	D12882	000558	0764706	8383 C	E1		
95	D12149	000558	0859332	8493 C	E1		
95	D15678	000558	0828044	8776 C	E1		
54	CMD105	000558	0949251	9195 C	E1		
49	CMD001	000558	0942885	8331 C	E1		
49	CMD105	000558	0838848	8880 C	E1		
49	D17478	000558	0564224	7601 C	E1		
42	D18477	000558	0564395	7950 C	E1		
50	D18982	000558	0795089	8530 C	E1		
3	CMD658	000558	0773998	8656 C	E1		
40	D16778	000558	0856019	8906 C	E1		
49	CMD116	000558	0880738	7334 C	E1		
90	CMD104	000558	0880733	8089 C	E1		
148	CMD101	000558	0880708	8232 C	E1		
10	D19919	000558	0933902	8234 C	E1		
90	CMD107	000558	0828040	8449 C	E1		
70	D18038	000558	0625842	8896 C	E1		
60	D12682	000558	0764705	8330 C	E1		
60	D70279	000558	0564238	7576 C	E1		
44	D12685	000558	0764709	8377 C	E1		
39	D15702	000558	0879460	8779 C	E1		
80	D28971	000558	0564202	7170 C	E1		
49	D18979	000558	0874038	7521 C	E1		

Generate SAP Number

Generate Catalog Number

Figura 80: Generación de SAP Numbers

Como se pudo observar en la Figura 80, todos los materiales ingresados cumplen con la característica de tener un estatus adecuado para continuar. A continuación, se comenzará con el ciclo para poder generar la carga de datos en la transacción.

Primeramente, se abrirá Power Automate, y de inmediato se mostrada la pantalla con el flujo para realizar el proceso, como se muestra en la Figura 81.

Power Automate Reyes, Juan Manuel ND/ASM-...

+ Nuevo flujo ▶ Ejecutar □ Detener ✎ Editar ... soft.com (default) (...) Configuración ? Ayuda 🔍 Buscar flujos

Inicio Mis flujos Compartidos conmigo Ejemplos Actualizar

Nombre	Modificado ↓	Estado
SAP - Inbound Number...	Hace < 1 minuto	No está en ejecución

Figura 81: Power Automate - flujo

A continuación, se dará clic en el botón de Ejecutar, y comenzará el proceso automatizado.

Primero que todo, se abrirá la transacción de SAP, que tiene la interfaz mostrada en la Figura 82.

Figura 82: Transaccion Inbound Numbers SAP

Como se puede observar en Figura 83, se realiza de forma automática la carga de la información, primeramente, con los datos que se mantendrán como constantes, y a continuación con los números de material y su cantidad.

Wa...	cantidad de entrada	ca.pendiente	Ord.compra	ltm	LiefCharge	C...	Inf...	data	FechaEntrg...	articl.proveedc...
	1.318,000	1.320,000	00356	0				25.01.24	25.01.2024	8804 D1
		1.500,000	00359	20				27.02.24	27.02.2024	8804 D1
		1.200,000	0035	20				26.03.24	26.03.2024	8804 D1

Figura 83: Proceso carga de información en la transacción

También se observa en la Figura 87, dentro de la aplicación, que en la columna nombrada como “Status”, todas las filas tienen la cantidad esperada por la orden, y no se encuentran coloreadas de ningún color ni marcan que no coincide la orden de compra, lo que indica que no se encontró ningún problema al realizar la carga de la información y todo el proceso se realizó correctamente.

Figura 87: Proceso finalizado en aplicación

Por lo que al trabajador solo le queda presionar el botón “Crear entrega entrante” dentro de la transacción (mostrado en la Figura 88), para obtener finalmente el número del Inbound necesario para culminar con esta tarea.

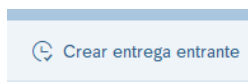
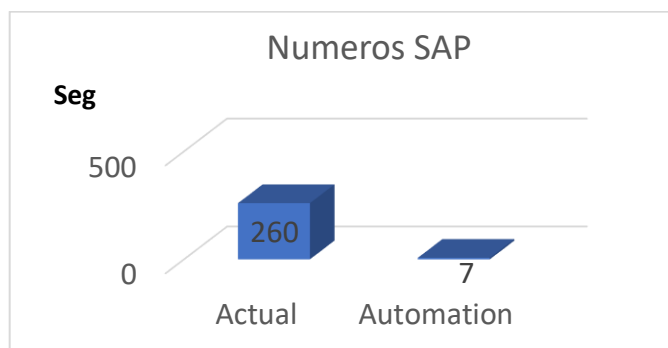


Figura 88: Crear Entrega entrante

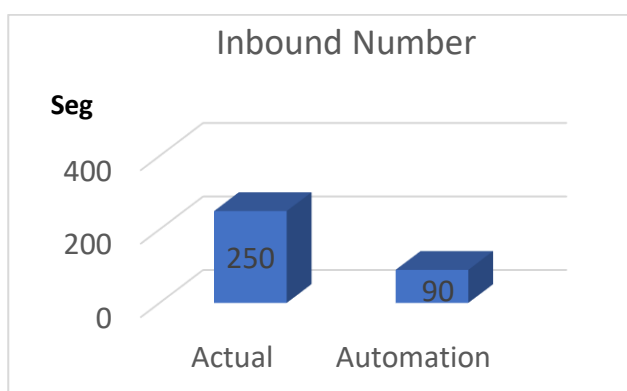
Ahora ya con la aplicación implementada es posible realizar una comparativa entre el proceso antes y después de ser automatizado.

Primeramente, analizando el tiempo invertido al realizar el proceso de obtener los números de SAP a partir de los números de catálogo (Para esta muestra se han definido 10 números de material distintos), en la Grafica 1 se observa el comportamiento del tiempo en el eje Y, y el eje X muestra el proceso actual contra la automatización, se observa que el trabajador promedio tardaba alrededor de 260 segundos en realizar esta tarea, y con la automatización este tiempo se reduce drásticamente a únicamente 7 segundos.



Grafica 1: Números SAP

Al analizar el tiempo invertido al realizar la carga de los números en la transacción, (Se ha decidió definir la muestra con 10 números de material distinto). En la Grafica 2 mostrada a continuación, se observa en el eje Y, el comportamiento del tiempo en segundos, y en el eje X se muestra el proceso actual contra la automatización. Se tiene que anteriormente el trabajador tardaba alrededor de 25 segundos en realizar la carga de un solo material (considerando que el trabajador realizaba la tarea de manera ardua y dedicada completamente a esta actividad), por lo que para realizar la carga de 10 números utilizaba aproximadamente 250 segundos. Con la automatización implementada este proceso se redujo drásticamente, ya que el tiempo para realizar la carga por material es únicamente de 9 segundos (dependiendo de la capacidad del computador, pudiendo ser incluso más rápido con una computadora más veloz), por lo que el tiempo para realizar la carga de 10 materiales sería de aproximadamente 90 segundos.



Grafica 2: Proceso Inbound

Traduciendo estos datos sobre la carga de los materiales en la transacción a números reales, se tiene que un solo trabajador (existen 4 trabajadores realizando esta labor), que mensualmente debe realizar la carga aproximada de 12471 números de material, estaría utilizando alrededor de 86.6 horas de su tiempo realizando esta tarea, y con la automatización, asumiendo un tiempo de 9 segundos por material, el tiempo invertido en el mes sería de sólo 31.8 horas, lo que representa un aumento del 63,2% en la eficiencia.

Recalcando que el trabajador puede realizar otras labores mientras se ejecuta la automatización, por lo que el tiempo invertido es mucho menor, y sería únicamente el tiempo en que realiza la carga de la información proveniente de las facturas para colocarlo en la aplicación, es decir únicamente minutos o si acaso un par de horas por mes.

Además, que al implementar la automatización se reducen al 100% los posibles errores manuales al cargar la información.

Esta libertad de tareas manuales permite a los empleados centrarse en actividades estratégicas, mejorando la productividad y potencialmente generando ingresos adicionales para la empresa.

CONCLUSIONES

Los resultados alcanzados en el proyecto han sido significativos y alineados con los objetivos iniciales propuestos. La implementación de la aplicación desarrollada para facilitar el proceso de generación de los "Inbound Numbers" ha sido exitosa y constituye un hito clave en la automatización de la cadena de suministro en Schaeffler. Esta aplicación, construida con Visual Basic for Applications (VBA) y en Power Automate, ha demostrado su eficacia al integrarse de manera efectiva con el entorno SAP ERP 6.0, logrando así optimizar y agilizar la generación de los Inbound Numbers de forma precisa y eficiente.

La importancia de estos resultados radica en la mejora significativa en la eficiencia operativa. La generación automatizada de los Inbound Numbers, ha permitido reducir considerablemente los tiempos de procesamiento, eliminando errores asociados a tareas manuales y mejorando la exactitud de los registros. Este logro no solo representa un avance en términos de eficiencia, sino que también contribuye a una mejor gestión de la información y una respuesta más ágil a los procesos internos de la empresa.

Las interfaces de usuario intuitivas desarrolladas para esta aplicación juegan un papel fundamental en la aceptación y adopción por parte del equipo operativo. La facilidad de uso y la comprensibilidad de las interfaces contribuyen significativamente a la efectividad de la herramienta, permitiendo a los usuarios interactuar de manera fluida y realizar sus tareas con mayor rapidez y precisión.

La integración exitosa abre un amplio abanico de posibilidades para la automatización de tareas repetitivas. La capacidad de generar scripts y programaciones específicas ha demostrado ser un recurso valioso para abordar procesos altamente repetitivos dentro del sistema, lo que ofrece un potencial enorme para optimizar otras áreas de la empresa y, en última instancia, mejorar la productividad general de Schaeffler.

En resumen, los resultados obtenidos a través de la implementación de esta aplicación desarrollada en VBA y Power Automate para la generación automatizada de Inbound Numbers han sido altamente satisfactorios y representan un paso significativo hacia la optimización de procesos en Schaeffler. La eficiencia mejorada, la reducción de errores y la capacidad de automatizar tareas repetitivas destacan el impacto positivo de esta solución en la mejora operativa y la competitividad de la empresa en un entorno empresarial dinámico y exigente.

OBJETIVOS EDUCACIONALES

A lo largo del desarrollo del proyecto, se evidenció el fortalecimiento y aplicación de diversas competencias alineadas con el perfil de egreso de la Ingeniería en Mecatrónica. La integración de conocimientos sólidos en áreas como la automatización o la programación fue esencial para la concepción y ejecución de la aplicación de automatización destinada a la generación de Inbound Numbers en Schaeffler. Este proceso permitió demostrar sólidos conocimientos en estas áreas, trabajando directamente en la implementación de soluciones para optimizar procesos.

La especialización en áreas específicas de la Ingeniería Mecatrónica fue clave para abordar con profundidad las necesidades del proyecto. El diseño y desarrollo de la aplicación en Visual Basic for Applications (VBA) y Power Automate, implicó un conocimiento especializado en la integración de sistemas, demostrando habilidades avanzadas en esta área de especialización.

La participación en equipos multidisciplinarios fue un aspecto crucial durante el proyecto. Colaborar con diversas áreas dentro de la empresa, incluyendo el área del manejo de materiales y otros sectores, permitió integrar distintas perspectivas y conocimientos, siendo esencial para el éxito de la implementación de la solución de automatización.

Además, al alcanzar los objetivos del proyecto, se puso de manifiesto la capacidad para la resolución de problemas de Ingeniería Mecatrónica. El análisis, diseño y ejecución de la solución para la generación automatizada de Inbound Numbers reflejó la habilidad para formular y resolver problemas complejos mediante la aplicación de conocimientos teóricos en contextos prácticos.

La comunicación efectiva con diversas audiencias, tanto de manera oral como a través de documentación escrita, fue esencial para explicar el diseño, desarrollo y funcionalidad de la aplicación. Esta habilidad de comunicación facilitó la comprensión y aceptación de la solución por parte del equipo operativo.

En resumen, el proyecto permitió al profesional aplicar y desarrollar competencias clave del perfil de egreso de la Ingeniería en Mecatrónica. Desde la resolución de problemas complejos hasta la comunicación efectiva y el trabajo en equipos multidisciplinarios, todo enmarcado en el desarrollo exitoso de una solución de automatización que impacta positivamente en los procesos de Schaeffler.

FUENTES DE INFORMACIÓN

Alexander, M., & Kusleika, D. (2017). *Excel VBA Programming for Dummies. For Dummies*.

Bowersox, D. J., Closs, D. J., & Cooper, M. B. (2007). *Supply Chain Logistics Management*. McGraw-Hill Education.

Chen, Q., Zhang, X., & Zhang, X. (2019). Intelligent Workflow Automation: A Comprehensive Review. *Journal of Network and Computer Applications*, 131, 64-80.

Davenport, T. H. (2018). *The AI Advantage: How to Put the Artificial Intelligence Revolution to Work*. MIT Press.

Fico, D. (2020). *Mendix Quick Start Guide: Rapid application development for web and mobile apps with Mendix*. Packt Publishing.

Laudon, K. C., & Laudon, J. P. (2016). *Management Information Systems: Managing the Digital Firm*. Pearson.

Manyika, J., Chui, M., Miremadi, M., Bughin, J., George, K., Willmott, P., & Dewhurst, M. (2017). *A Future That Works: Automation, Employment, and Productivity*. McKinsey Global Institute. Retrieved from <https://www.mckinsey.com/~media/mckinsey/featured%20insights/digital%20disruption/ai%20automation%20and%20the%20future%20of%20work/mgi-a-future-that-works-executive-summary.ashx>

Monczka, R. M., Handfield, R. B., Giunipero, L. C., & Patterson, J. L. (2018). *Purchasing and Supply Chain Management*. Cengage Learning.

Santos, A. L., Tribolet, J. M., & Oliveira, T. (2020). Business Process Management and Automation with Low-Code/No-Code Platforms: A Survey of Practitioners' Perceptions. *Information Systems*, 92, 101532.

SAP AG. (2019). *SAP ERP 6.0: User Guide*. SAP Press.

Shang, S., & Seddon, P. B. (2018). A Comprehensive Review of Enterprise Resource Planning (ERP) Implementations in Different Countries. *Journal of Global Information Management*, 26(4), 1-25.

Van der Meer, R. (2012). *Accounts Payable and Sarbanes-Oxley: Strengthening Your Internal Controls*. John Wiley & Sons.

FUENTES DE INFORMACION ILUSTRACIONES

Figura 1: Sistemas ERP. (2023, 24 de octubre). CEGRESADOS UDLAP. <https://cegresados.udlap.mx/wp-content/uploads/2023/05/ERP.jpg>

Figura 2: Ventana SAP. (2023). En Einstieg in ABAP. Rheinwerk Verlag. https://openbook.rheinwerk-verlag.de/einstieg_in_abap/bilder/01_004.png

Figura 3: Inbound. (2023, 1 de enero). En Trimble Geospatial. Trimble. <https://tl.trimble.com/wp-content/uploads/2023/01/2.png>

Figura 4: Órdenes de Compra. (Sin fecha). En Website Files. https://assets-global.website-files.com/575ef60509a5a7a9116d9f8c/5de943dfefccd85744d492d8_order-compra.png

Figura 5: Facturas de Compra. (2022, septiembre). En Mis Cuentas. https://sp-ao.shortpixel.ai/client/q_lossless,ret_img,w_1280/https://www.miskuentas.com/wp-content/uploads/2022/09/9ec13198-3b73-4fde-a96d-5d403a8f74ae.jpg

Figura 6: Pantalla Mendix. (2023, 25 de julio). En Mendix. Mendix. <https://www.mendix.com/wp-content/uploads/footer-studio-e1687874689701-1536x980.png>

Figura 7: Ventana Power Automate: (Sin fecha). En Microsoft Power Automate. Microsoft. <https://learn.microsoft.com/es-es/power-automate/media/getting-started/home-full-numbers.png>

Figura 8: Ventana VBA. (Sin fecha). En Microsoft Office VBA. Microsoft. <https://learn.microsoft.com/en-us/office/vba/images/5d9acd78-5168-4a0c-83b6-3f1e440bf649.png>

Figura 9. Diagrama de Operaciones. Figura de autoría Propia

Figura 10. Automatizar Operaciones en SAP. Figura de autoría Propia

Figura 11. Ejemplo de Factura. Figura de autoría Propia

Figura 12. Interfaz 1. Figura de autoría Propia

Figura 13. Variables Programa. Figura de autoría Propia

Figura 14: Pedir nombre de archivo. Figura de autoría Propia

Figura 15: InputBox. Figura de autoría Propia

Figura 16: Error con el nombre de archivo. Figura de autoría Propia

Figura 17: Asignación de columnas. Figura de autoría Propia

Figura 18: Manejo de error tabla vacía. Figura de autoría Propia

Figura 19: Ciclo For. Figura de autoría Propia

Figura 20: Comparación entre valores. Figura de autoría Propia

Figura 21: Asignación de valores. Figura de autoría Propia

Figura 22: Colorear Status. Figura de autoría Propia

Figura 23: Obtención de valores. Figura de autoría Propia

Figura 24: Asignación de Proveedor. Figura de autoría Propia

Figura 25: Interfaz del programa 2. Figura de autoría Propia

Figura 26: Comparativa de columnas. Figura de autoría Propia

Figura 27: Asignación de datos 2 caso. Figura de autoría Propia

Figura 28: Cierre de Código. Figura de autoría Propia

Figura 29: Variables proveedor. Figura de autoría Propia

Figura 30: Solicitar nombre de documento. Figura de autoría Propia

Figura 31: Asignación de valores con SAP Number. Figura de autoría Propia

Figura 32: Limpiar tabla. Figura de autoría Propia

Figura 33: Instrucciones. Figura de autoría Propia

Figura 34: Interfaz de usuario final. Figura de autoría Propia

Figura 35: Interfaz de Usuario SCOPE. Figura de autoría Propia

Figura 36: Variables SCOPE. Figura de autoría Propia

Figura 37: Solicitar nombre SCOPE. Figura de autoría Propia

Figura 38: Asignación de hojas SCOPE. Figura de autoría Propia

Figura 39: Asignación de variables a objetos. Figura de autoría Propia

Figura 40: Ciclo For SCOPE. Figura de autoría Propia

Figura 41: Numero de Catalogo SCOPE. Figura de autoría Propia

Figura 42: Dividir Cadena. Figura de autoría Propia

Figura 43: Asignación de valores SCOPE. Figura de autoría Propia

Figura 44: Asignación de valores SCOPE_supplier. Figura de autoría Propia

Figura 45: Mensaje de éxito SCOPE. Figura de autoría Propia

Figura 46: Variables y Nombre de libro SCOPE. Figura de autoría Propia

Figura 47: Establecer Objetos SCOPE. Figura de autoría Propia

Figura 48: Asignación de Valores SCOPE con SAP Numbers. Figura de autoría Propia

Figura 49: Limpiar Tabla SCOPE. Figura de autoría Propia

Figura 50: UserForm. Figura de autoría Propia

Figura 51: Abrir el libro. Figura de autoría Propia

Figura 52: Código UserForm. Figura de autoría Propia

Figura 53: Seleccionar Base de datos. Figura de autoría Propia

Figura 54: Iniciar Excel. Figura de autoría Propia

Figura 55: Cuadro De Mensaje. Figura de autoría Propia

Figura 56: Hoja De Trabajo Activa. Figura de autoría Propia

Figura 57: Variables Constantes. Figura de autoría Propia

Figura 58: Obtener Última Fila. Figura de autoría Propia

Figura 59: Almacenar Columnas. Figura de autoría Propia

Figura 60: Abrir Transacción. Figura de autoría Propia

Figura 61: Rellenar Campos En La Transacción. Figura de autoría Propia

Figura 62: Bucle Pa. Figura de autoría Propia

Figura 63: Rellenar Materiales. Figura de autoría Propia

Figura 64: Enviar Tecla. Figura de autoría Propia

Figura 65: Almacenar Órdenes De Compra. Figura de autoría Propia

Figura 66: Condicional Órdenes De Compra. Figura de autoría Propia

Figura 67: Modificar Cantidad. Figura de autoría Propia

Figura 68: Clic De Selección. Figura de autoría Propia

Figura 69: Botón Adopt Items. Figura de autoría Propia

Figura 70: Escribir Cantidades. Figura de autoría Propia

Figura 71: Condicionales Órdenes De Compra. Figura de autoría Propia

Figura 72: Condicional Else. Figura de autoría Propia

Figura 73: Final Del Bucle. Figura de autoría Propia

Figura 74: Formulario Selección. Figura de autoría Propia

Figura 75: Interfaz De Usuario Scope. Figura de autoría Propia

Figura 76: Interfaz De Usuario Mm Aval. Figura de autoría Propia

Figura 77: Factura Proveedor. Figura de autoría Propia

Figura 78: Plantilla Con Información Cargada. Figura de autoría Propia

Figura 79: Introducir Base De Datos. Figura de autoría Propia

Figura 80: Generación De SAP Numbers. Figura de autoría Propia

Figura 81: Power Automate – Flujo. Figura de autoría Propia

Figura 82: Transaccion Inbound Numbers Sap. Figura de autoría Propia

Figura 83: Proceso Carga De Información En La Transacción. Figura de autoría Propia

Figura 84: Comparativa Órdenes De Compra. Figura de autoría Propia

Figura 85: Adopt Items. Figura de autoría Propia

Figura 86: Transaccion final. Figura de autoría Propia

Figura 87: Proceso Finalizado En Aplicación. Figura de autoría Propia

Figura 88: Crear Entrega Entrante. Figura de autoría Propia

ANEXOS

SCHAEFFLER

Schaeffler México Servicios S de RL de CV.
Autopista Chamapa-Lechería km 2.5
Nave B-011 San Martín Obispo
Cuautitlán Izcalli, Estado de México
C.P. 54769
+5550626010

Ciudad de México a 23 de Enero del 2024

Estimados miembros del Tecnológico de Estudios Superiores de Coacalco:

Por este medio me dirijo a ustedes para compartir la experiencia que hemos vivido en **SCHAEFFLER AUTOMOTIVE AFTERMARKET MEXICO S. DE R.L. DE C.V.** con la incorporación de **Juan Manuel Reyes Rojas** número de matrícula **201920104** y número de empleado **32011070**, que actualmente se desempeña como trainee en el área de SCM & Operations en nuestra organización liderando el proyecto "Inbound RPA Process" dentro de nuestro programa **Semillero de talentos**.

Durante el periodo que Juan Manuel ha formado parte de nuestro equipo, ha demostrado gran dedicación con sus responsabilidades. Sus contribuciones significativas en diversos proyectos dentro de la empresa han destacado su capacidad para abordar desafíos complejos con ingenio y profesionalismo.

Juan Manuel ha participado activamente en iniciativas, presentando ideas que han impulsado la eficiencia operativa y la innovación en el área de Supply Chain. Su habilidad para colaborar efectivamente con colegas de diferentes departamentos ha sido un activo valioso, y su compromiso con la mejora continua ha dejado una impresión positiva en toda la organización.

En el tiempo en el que **Juan Manuel** ha colaborado para esta empresa, ha demostrado ser una persona profesional en sus actividades, cumpliendo metas y objetivos planteados, por lo que no tenemos inconveniente en extender la presente recomendación de obtener el grado académico de **Ingeniero en Mecatrónica**. Creemos en la aportación y la experiencia que ha obtenido en la práctica dentro del entorno profesional, que es la finalidad del programa y alineado a nuestros valores.

Guzman,
Eber
Atentamente

Firmado digitalmente por Guzman,
Eber
DN: cn=Guzman, Eber, o=DE,
ou=Schaeffler Group
Fecha: 2024.01.23 12:20:29 -0600

Eber Guzman Rodriguez
SCM & Operations Manager Americas Central

**Schaeffler Automotive
Aftermarket Mexico,
S. de R.L. de C.V.
Recursos Humanos
Registro Patronal M4932080100
RFC SAA970102TH8**