

**UNIVERSIDAD MAYOR DE SAN ANDRÉS**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA INDUSTRIAL**



**MEMORIA LABORAL**

**AUTOMATIZACIÓN DEL FORMULARIO PARA EL PROCEDIMIENTO DE  
GESTIÓN DE RIESGOS PARA EL ALTA O MODIFICACIÓN DE ROLES DE  
USUARIOS DEL SISTEMA SAP EN SOBOCE S.A.**

**Memoria Laboral presentada para la obtención del grado de Licenciatura**

**POR: CORDERO TINTAYA JUAN VIDAL**

**TUTOR: Ing. CARLA KAUNE SARABIA**

**LA PAZ – BOLIVIA**

**2023**



**UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE INGENIERIA**



**LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.**

**LICENCIA DE USO**

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la cita o referencia correspondiente en apego a las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

**TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADAS EN LA LEY DE DERECHOS DE AUTOR.**

UNIVERSIDAD MAYOR DE SAN ANDRÉS  
FACULTAD DE INGENIERÍA  
INGENIERÍA INDUSTRIAL

Memoria laboral:

AUTOMATIZACIÓN DEL FORMULARIO PARA EL PROCEDIMIENTO DE  
GESTIÓN DE RIESGOS PARA EL ALTA O MODIFICACIÓN DE ROLES DE  
USUARIOS DEL SISTEMA SAP EN SOBOCE S.A.

Presentado por:

Univ. Cordero Tintaya Juan Vidal

Para optar por el grado académico de licenciatura en ingeniería industrial

Nota numeral: .....

Nota literal: .....

Ha sido: .....

Director de Carrera de ingeniería industrial:

Ing. M.Sc. Franz Zenteno Benítez .....

Tutor:

Ing. Carla Kaune Sarabia .....

Tribunales:

Ing. Lucio Grover Sánchez Eid .....

Ing. Edgar Quiroga Vargas .....

Ing. Mario Zenteno Benítez .....

*“Porque Jehová da la sabiduría, y de su boca viene el conocimiento y la inteligencia.”*

**Proverbios 2:6**

## RESUMEN

En la actualidad, la automatización de procesos y el desarrollo de software desempeñan un papel fundamental en el mundo empresarial. Con el avance de la tecnología de la información, se ha vuelto esencial contar con sistemas eficientes y herramientas adecuadas para optimizar y agilizar los procesos organizacionales.

En la parte introductoria se muestran los antecedentes laborales del autor y el caso de estudio, para el que se realiza el análisis del problema y los objetivos propuestos para implementar la automatización del formulario para el procedimiento de gestión de riesgos para el alta o modificación de roles de usuarios del sistema SAP en SOBOCE S.A.

En el marco teórico se abordan conceptos como la automatización, desarrollo de software, sistemas de información, herramientas específicas y su impacto en los procesos de negocio. Esto proporciona la base teórica necesaria para comprender la implementación del formulario automatizado.

El diagnóstico realizado permite identificar la ineficiencia en el llenado del formulario antes de la implementación de la automatización. Como resultado del diagnóstico, la propuesta se centra en el diseño de la arquitectura, la interfaz y los algoritmos necesarios para el funcionamiento del formulario.

El marco aplicativo describe la construcción e implementación del formulario abordando aspectos como la codificación realizada utilizando el lenguaje VBA y se presentan las pruebas correspondientes para verificar su efectividad.

La evaluación técnico-financiera compara los resultados obtenidos antes y después de la implementación del formulario automatizado. Estos resultados demuestran los beneficios de la implementación del formulario automatizado en términos de eficiencia y calidad del software, lo que indica que la solución propuesta ha sido exitosa.

Palabras clave: Automatización, Desarrollo de software, Sistemas de información, Herramientas específicas, Algoritmos, Codificación, Lenguaje VBA.

## SUMMARY

Currently, process automation and software development play a fundamental role in the business world. With the advancement of information technology, it has become essential to have efficient systems and appropriate tools to optimize and streamline organizational processes.

The introductory part presents the author's work background and the case study for which the analysis of the problem and proposed objectives for implementing the automation of the form for the risk management procedure for adding or modifying user roles in the SAP system at SOBOCE S.A. are carried out.

The theoretical framework addresses concepts such as automation, software development, information systems, specific tools, and their impact on business processes. This provides the necessary theoretical basis to understand the implementation of the automated form.

The conducted diagnosis allows identifying inefficiency in the form filling process before the automation implementation. As a result of the diagnosis, the proposal focuses on the design of the architecture, interface, and algorithms necessary for the form's operation.

The applicative framework describes the construction and implementation of the form, addressing aspects such as coding using the VBA language and presenting the corresponding tests to verify its effectiveness.

The technical and financial evaluation compares the results obtained before and after the implementation of the automated form. These results demonstrate the benefits of implementing the automated form in terms of software efficiency and quality, indicating that the proposed solution has been successful.

**Keywords:** Automation, Software development, Information systems, Specific tools, Algorithms, Coding, VBA language.

## Contenido

Área I - Generalidades de la actividad profesional .....	1
Área II - Caso de estudio .....	6
Capítulo 1: Introducción .....	7
1.1. Planteamiento del problema .....	7
1.1.1. Análisis causa-efecto .....	7
1.1.2. Problema.....	9
1.2. Objetivo general .....	9
1.3. Objetivos específicos.....	9
1.4. Justificación .....	9
1.4.1. Justificación Económico-Social .....	9
1.4.2. Justificación académica .....	9
Capítulo 2: Marco Teórico.....	11
2.1. Automatización.....	11
2.2. Software.....	12
2.2.1. Desarrollo de software.....	12
2.2.2. Modelo.....	13
2.2.2.1. Prototipo.....	15
2.2.3. Arquitectura de software .....	15
2.2.4. Lenguaje de Programación .....	17
2.2.5. Algoritmo.....	17
2.2.6. Ingeniería de requerimientos .....	18
2.2.7. Calidad del software .....	19
2.3. Sistema.....	20
2.3.1. Proceso.....	21
2.3.2. Enfoque en procesos.....	21
2.3.3. Procedimiento.....	22
2.3.4. Sistema de información .....	22
2.3.4.1. Componentes de un sistema de información.....	22
2.3.4.2. Tipos de sistemas de información .....	24
2.3.4.3. Organización .....	24
2.3.4.4. Tecnología de la información.....	25
2.3.4.5. Sistemas de información en los procesos de negocio .....	25
2.3.4.6. Base de datos.....	26
2.3.4.7. Modelo relacional.....	27
2.4. Herramientas específicas .....	28
2.4.1. Microsoft Excel .....	28

2.4.2.	VBA.....	29
2.4.3.	SQL.....	31
2.4.4.	SAP.....	32
2.5.	Personalización.....	33
2.6.	Ética en el desarrollo de software.....	35
Capítulo 3:	Diagnóstico y Propuesta.....	36
3.1.	Diagnóstico.....	36
3.1.1.	Situación de la empresa.....	36
3.1.1.1.	Lineamiento para el desarrollo de software.....	37
3.1.2.	Identificación de la información del proceso.....	38
3.1.2.1.	Diagrama de flujo de funciones cruzadas.....	38
3.1.2.2.	Diagrama de flujo del proceso.....	40
3.2.	Propuesta.....	42
3.2.1.	Requerimientos.....	43
Capítulo 4:	Marco Aplicativo.....	45
4.1.	Diseño.....	45
4.1.1.	Arquitectura.....	45
4.1.2.	Interfaz.....	47
4.1.3.	Algoritmos.....	48
4.2.	Construcción.....	49
4.2.1.	Código.....	49
4.2.2.	Entorno de desarrollo integrado.....	50
4.2.3.	Codificación.....	51
4.3.	Pruebas.....	51
4.3.1.	Copia de datos.....	52
4.3.2.	Transferencia de datos al formato del formulario.....	54
4.3.3.	Creación de copia del formulario.....	55
4.3.4.	Transferencia de datos al formato de registro.....	55
4.3.5.	Registro en la Base de Datos.....	57
4.3.6.	Modificación de registros.....	58
4.3.7.	Generación de reporte.....	59
Capítulo 5:	Evaluación técnico-financiera.....	61
5.1.	Comparación de procedimientos antes y después de la implementación.....	61
5.2.	Evaluación de la calidad del software.....	63
5.3.	Análisis costo-beneficio.....	64
5.4.	Evaluación de impacto.....	66
Capítulo 6:	Conclusiones y recomendaciones.....	68
6.1.	Conclusiones.....	68



6.2. Recomendaciones .....	69
Área III – Análisis de la actividad laboral .....	70
Bibliografía .....	71
ANEXOS .....	73
Anexo 1: Sintaxis del lenguaje VBA .....	73
Anexo 2: Formulario.....	76
Anexo 3: Diagramas de flujo .....	78
Anexo 4: Descripción del código.....	86
Anexo 5: Atributos de evaluación de la calidad del software.....	102
Anexo 6: Certificados de trabajo .....	105
Índice de tablas .....	iv
Índice de imágenes.....	vii

## Índice de tablas

Tabla 1: Descripción de la actividad laboral.....	2
Tabla 2: Categorías del diagrama de Ishikawa .....	8
Tabla 3: Conocimientos adquiridos y aplicados .....	10
Tabla 4: Categorías de Software .....	12
Tabla 5: Actividades estructurales del desarrollo de software.....	13
Tabla 6: Modelos para el desarrollo de software .....	14
Tabla 7: Estilos de arquitectura de software .....	16
Tabla 8: Tareas de la ingeniería de requerimientos .....	18
Tabla 9: Elementos de la calidad de software.....	20
Tabla 10: Componentes de un sistema de información .....	23
Tabla 11: Tipos de sistemas de información.....	24
Tabla 12: Terminología sobre base de datos.....	26
Tabla 13: Características de Microsoft Excel .....	28
Tabla 14: Características del lenguaje VBA.....	29
Tabla 15: Procedimientos del lenguaje VBA.....	30
Tabla 16: Instrucciones del lenguaje VBA .....	30
Tabla 17: Tipos de dato.....	31
Tabla 18: Características de SQL .....	31
Tabla 19: Términos del sistema SAP.....	33
Tabla 20: Comparación de la personalización con VBA y Visual Studio .....	34
Tabla 21: Costo de desarrolladores de soluciones SAP .....	37
Tabla 22: Símbolos del diagrama de flujo del proceso.....	40
Tabla 23: Requerimientos de desarrollo .....	43
Tabla 24: Requerimientos técnicos.....	44
Tabla 25: Arquitectura de capas del formulario automatizado .....	46
Tabla 26: Símbolos de un diagrama de flujo .....	49
Tabla 27: Técnicas de código limpio.....	50
Tabla 28: Características del editor de código de Microsoft Office .....	51
Tabla 29: Parámetros de la prueba de copia de datos. ....	52

Tabla 30: Parámetros de la prueba de transferencia de datos .....	54
Tabla 31: Parámetros de la prueba de creación de copia .....	55
Tabla 32: Parámetros de la prueba de transferencia de datos al formato de registro.....	56
Tabla 33: Parámetros de la prueba de registro en base de datos.....	57
Tabla 34: Parámetros de la prueba de modificación de registros .....	58
Tabla 35: Parámetros de la prueba de generación de reporte .....	59
Tabla 36: Comparación de procedimientos antes y después de la implementación .....	62
Tabla 37: Cumplimiento de las características de calidad.....	64
Tabla 38: Costos del proyecto.....	65
Tabla 39: Métricas de evaluación .....	66
Tabla 40: Sintaxis del lenguaje VBA.....	73
Tabla 41: Operadores.....	75
Tabla 42: Datos del formulario.....	76
Tabla 43: Variables.....	86
Tabla 44: Variables públicas.....	87
Tabla 45: Constantes.....	87
Tabla 46: Formato de colores para la descripción del código .....	87
Tabla 47: Código para la lista de archivos en la hoja “COPIA” .....	87
Tabla 48: Código para la lista de hojas en la hoja “COPIA” .....	88
Tabla 49: Código para ejecutar el procedimiento “actualizar_formulario” .....	88
Tabla 50: Código para armar la consulta del reporte.....	88
Tabla 51: Código para la copia de datos.....	89
Tabla 52: Código para transferir datos al formato de la hoja “FORMULARIO” .....	90
Tabla 53: Código para crear un copia del formulario .....	91
Tabla 54: Código para transferir datos al formato de la hoja “REGISTRO” .....	92
Tabla 55: Código para insertar registros.....	93
Tabla 56: Código para modificar registros .....	94
Tabla 57: Código para generar un reporte .....	95
Tabla 58: Código de conexión al servidor .....	95
Tabla 59: Código con la consulta para seleccionar datos .....	96

Tabla 60: Código con la consulta para insertar registros .....	96
Tabla 61: Código con la consulta para modificar registros .....	96
Tabla 62: Código con la consulta para generar un reporte .....	96
Tabla 63: Código para convertir el formato de fecha .....	96
Tabla 64: Código para obtener datos .....	97
Tabla 65: Código para actualizar la hoja “FORMULARIO” .....	97
Tabla 66: Código para actualizar funciones.....	97
Tabla 67: Código para actualizar listas desplegables .....	97
Tabla 68: Código para mostrar mensaje de error (Nombre de archivo) .....	98
Tabla 69: Código para mostrar mensaje de error (Nombre de Hoja) .....	98
Tabla 70: Código para mostrar mensaje de error (Conexión).....	98
Tabla 71: Código para mostrar mensaje de error (Descripción).....	98
Tabla 72: Código para mostrar mensaje de advertencia (Sentencia Select) .....	98
Tabla 73: Código para mostrar mensaje informativo (Registros insertados) .....	98
Tabla 74: Código para mostrar mensaje informativo (Registros modificados) .....	99
Tabla 75: Código para mostrar mensaje informativo (Registro existente) .....	99
Tabla 76: Código para mostrar mensaje informativo (Registro inexistente) .....	99
Tabla 77: Código para mostrar mensaje informativo (Registro finalizado) .....	99
Tabla 78: Código para borrar datos .....	99
Tabla 79: Código para revisar formato .....	100
Tabla 80: Código para marcar el color de fondo de un rango .....	100
Tabla 81: Código para guardar archivo .....	101
Tabla 82: Código para editar rangos .....	101
Tabla 83: Código que devuelve el número de datos en un rango .....	101
Tabla 84: Código que devuelve el número de última fila.....	101
Tabla 85: Código para pegar datos .....	101
Tabla 86: Atributos de Funcionalidad.....	102
Tabla 87: Atributos de Confiabilidad .....	102
Tabla 88: Atributos de Usabilidad .....	103

## Índice de imágenes

Imagen 1: Organigrama de SOBOCE S.A.....	4
Imagen 2: Organigrama de PRO-BOLIVIA.....	5
Imagen 3: Diagrama causa-efecto (diagrama de Ishikawa).....	8
Imagen 4: Esquema de un sistema de información.....	23
Imagen 5: Representación del modelo relacional.....	27
Imagen 6: Diagrama de flujo de funciones cruzadas.....	39
Imagen 7: Diagrama de flujo del proceso sin la automatización.....	41
Imagen 8: Diagrama de flujo del proceso propuesto.....	42
Imagen 9: Capas del formulario automatizado.....	46
Imagen 10: Sección de datos generales.....	47
Imagen 11: Sección de usuarios.....	47
Imagen 12: Sección de riesgos.....	48
Imagen 13: Sección de aprobadores.....	48
Imagen 14: Conjunto de datos de prueba.....	52
Imagen 15: Configuración.....	53
Imagen 16: Selección del nombre de archivo.....	53
Imagen 17: Selección del nombre de hoja.....	54
Imagen 18: Ejecución del procedimiento A_COPIAR_DATOS.....	54
Imagen 19: Ejecución del procedimiento B_FORMATO_FORMULARIO.....	55
Imagen 20: Ejecución del procedimiento C_CREAR_COPIA.....	55
Imagen 21: Selección nombre de formulario.....	56
Imagen 22: Ejecución del procedimiento D_FORMATO_REGISTRO.....	56
Imagen 23: Ejecución del procedimiento E_REGISTRO_BD.....	57
Imagen 24: Mensaje informativo de registros insertados.....	57
Imagen 25: Ejecución del procedimiento F_MODIFICAR_BD.....	58
Imagen 26: Mensaje informativo de registros modificados.....	58
Imagen 27: Selección de campo.....	59
Imagen 28: Ejecución del procedimiento G_REPORTE.....	60

Imagen 29: Diagrama de flujo del algoritmo de copia de datos .....	78
Imagen 30: Diagrama de flujo del algoritmo de revisión .....	79
Imagen 31: Diagrama de flujo del algoritmo de formato .....	80
Imagen 32: Diagrama de flujo del algoritmo de copia de un archivo.....	81
Imagen 33: Diagrama de flujo del algoritmo de transferencia de datos .....	82
Imagen 34: Diagrama de flujo del algoritmo de consulta SQL (Registro) .....	83
Imagen 35: Diagrama de flujo del algoritmo de consulta SQL (Modificación).....	84
Imagen 36: Diagrama de flujo del algoritmo de consulta SQL (Reporte).....	85

## Área I - Generalidades de la actividad profesional

Durante su trayectoria profesional, el autor tuvo la oportunidad de desempeñar tareas relacionadas con en el análisis de procesos y el manejo de grandes volúmenes de datos en reconocidas organizaciones como SOBOCE S.A. y PRO-BOLIVIA.

SOBOCE S.A. (Sociedad Boliviana de Cemento) es una reconocida empresa cementera fundada en 1925 en La Paz que cuenta con tres plantas productoras de cemento, doce plantas de hormigón pre mezclado y dos plantas de producción de agregados pétreos. Es la empresa líder en la producción y comercialización de cemento, hormigón pre mezclado, áridos, prefabricados y servicios para el sector de la construcción en Bolivia.

PRO-BOLIVIA, por otro lado, es una entidad desconcentrada del Ministerio de Desarrollo Productivo y Economía Plural, creada en 2008 para brindar servicios de desarrollo empresarial a organizaciones económicas como la micro y pequeña empresa.

La experiencia adquirida en estas organizaciones ha permitido al autor adquirir habilidades de programación para el desarrollo de complementos en las aplicaciones de Microsoft Office para optimizar procesos. Además, ha desempeñado roles clave en la verificación y validación de datos técnicos en bases de datos, el relevamiento de información para documentar procesos y la elaboración de propuestas y procedimientos. También ha sido responsable de la recepción de materia prima, el control de inventarios y la elaboración de reportes de producción.

En la Tabla 1, se describe detalladamente la trayectoria laboral del autor, se destaca las actividades más relevantes en cada entidad. En el Anexo 6 se adjuntan las referencias de las experiencias laborales mencionadas para garantizar la veracidad de la información presentada en la memoria laboral y respaldar la credibilidad del autor en la realización del caso de estudio presentado.

**Tabla 1: Descripción de la actividad laboral**

<b>Cargo</b>	<b>Descripción</b>
<b>Consultor independiente</b> Mar. 2019 – Actualidad	Desarrollo de complementos para las aplicaciones de Microsoft Office. Los principales trabajos se realizaron para SOBOCE S.A. <ul style="list-style-type: none"><li>• Desarrollo de una herramienta de recálculo tributario</li><li>• Mejoras en el formulario automatizado “XRT-RCI.RP.001 FORMULARIO DE GESTIÓN DE RIESGOS PARA EL ALTA O MODIFICACIÓN DE ROLES”</li></ul>
<b>Consultor por producto PRO-BOLIVIA</b> Sept. 2019 - Mar. 2020	Encargado de la verificación y validación de datos técnicos declarados en la base de datos del programa PROMyPE y el sistema SISEGPRO en las regionales de El Alto y Oruro. Para reducir la posibilidad de error durante la revisión se desarrolló una herramienta de específica de Microsoft Excel.
<b>Analista de Procesos Junior</b> SOBOCE S.A. Dic. 2018 - Abr. 2019	Encargado del relevamiento de datos e información en los siguientes proyectos: <ul style="list-style-type: none"><li>• Sistema de facturación electrónica.</li><li>• Revisión del proceso de Alta, Baja y Modificación de Usuarios (referente a activos de TI). Durante este proyecto se presentó una propuesta de generación de formularios desde la plataforma de Service Desk con la finalidad de reemplazar los formularios físicos.</li><li>• Elaboración del procedimiento de “Gestión de riesgos para el alta o modificación de roles”. Durante este proyecto se presentó la propuesta de la primera versión del formulario automatizado.</li></ul>
<b>Analista del Centro de Información de Fábrica</b> SOBOCE S.A. May. 2018 - Oct. 2018	Responsable de la recepción de materia prima, control de inventarios, elaboración de reportes de producción en planta EMISA (Oruro). También se asumió la responsabilidad del registro y control de datos en el sistema SAP. Se desarrolló una herramienta para optimizar el proceso de elaboración de reportes, mediante la copia automática de datos desde los reportes de la balanza de pesaje, el laboratorio de pruebas, el área de mantenimiento y los registros de producción de la planta.
<b>Analista de Procesos Junior</b> SOBOCE S.A. Ene. 2018 - Abr. 2018	Encargado del relevamiento de datos e información para el proceso de pesaje de camiones en planta Viacha, con el objetivo de elaborar un documento de requerimientos para el desarrollo de un nuevo software para el pesaje de camiones.
<b>Pasante</b> SOBOCE S.A. Sep. 2017 - Oct. 2017	Apoyo en el proyecto “Gap Analysis” (Análisis de brechas), donde se realizó el relevamiento de datos en procesos diseñados y adoptados por la implementación del sistema SAP (HANA).

Fuente: Elaboración propia



## **SOBOCE S.A.**

SOBOCE S.A. produce cemento en varias plantas ubicadas en diferentes departamentos de Bolivia. A lo largo de su historia, ha tenido un importante crecimiento desde su fundación, consolidándose como líder en la industria del cemento en Bolivia. Ha experimentado cambios en la propiedad a lo largo de su historia, actualmente los accionistas principales de SOBOCE S.A. son YURA inversiones Bolivia S.A.

Las principales plantas de producción de SOBOCE S.A. son:

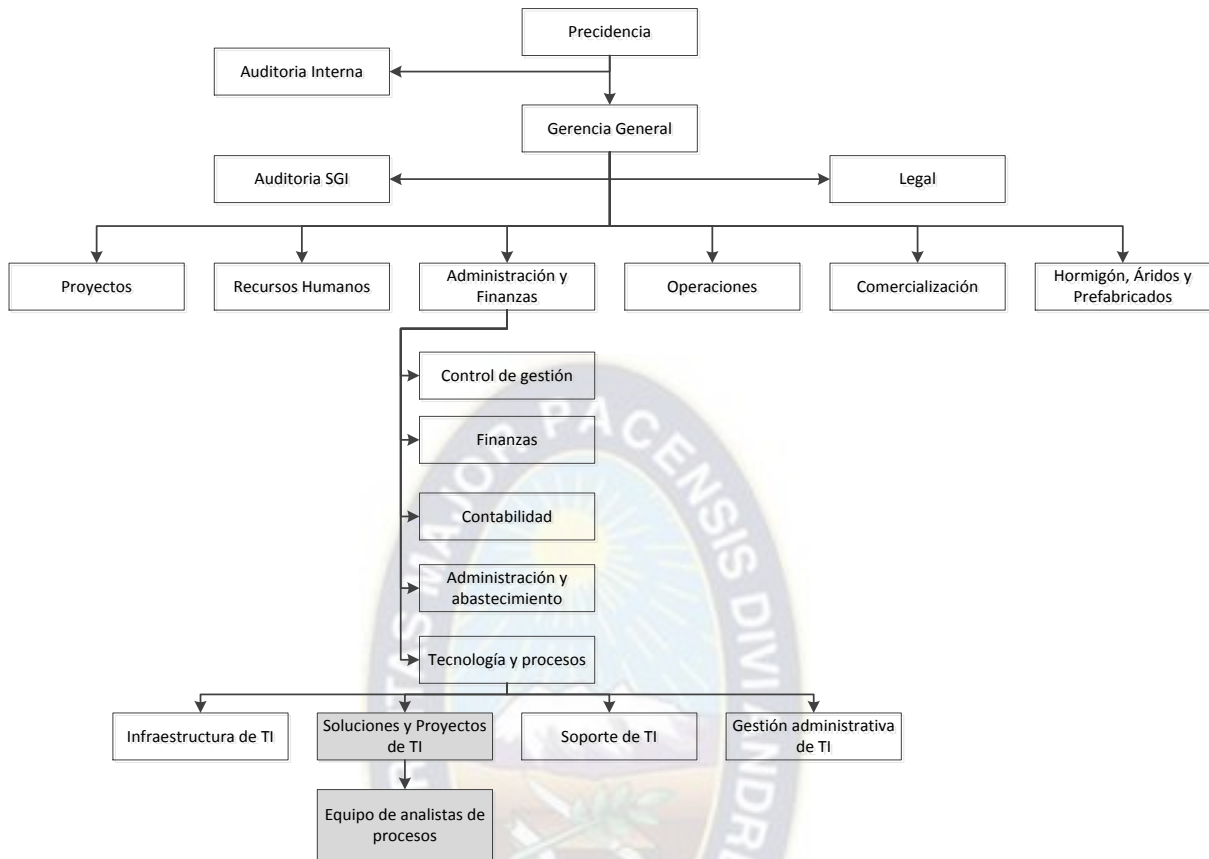
- Planta Industrial de Viacha, situada a 35 km. de la ciudad de La Paz y está constituida por tres líneas de producción de clinker y tres de cemento.
- Planta Warnes, ubicada a 25 kilómetros de la ciudad de Santa Cruz y está constituida por una terminal de molienda de cemento. El clinker es transportado desde la ciudad de La Paz y Tarija.
- Planta El Puente, ubicada a 91 kilómetros de la ciudad de Tarija, está constituida por dos líneas completas de producción de clinker y una de cemento.
- Plantas Ready Mix instaladas en las ciudades principales del eje troncal: La Paz (5), Cochabamba (1), Santa Cruz (1), Tarija (1) y Oruro (1). Además de proveer hormigón pre mezclado a las ciudades mencionadas, también provee hormigón prácticamente a nivel nacional gracias al uso de plantas dosificadoras móviles.
- Plantas de áridos en La Paz y Santa Cruz. Los productos más importantes de ambas plantas son la arena, grava, material para capa base, subbase y áridos para mezcla asfáltica.
- Plantas de prefabricados en La Paz y Santa Cruz. Los productos más importantes de ambas plantas son los pretensados, bloques y casetones.

## **Organización**

El organigrama de la empresa SOBOCE S.A. presenta una estructura jerárquica, en la que se pueden identificar los diferentes niveles de gestión y los departamentos correspondientes. A la cabeza de la presidencia de la junta directiva, debajo se encuentra la Gerencia General responsable de implementar estrategias y coordinar las operaciones diarias de la empresa. A continuación, se encuentran los departamentos, de Proyectos, Recursos Humanos, Administración y Finanzas, Operaciones, Comercialización y la Gerencia de Nacional de Hormigón, Áridos y Prefabricados.

Debido al tamaño de la empresa, en la Imagen 1 solo se desglosa a detalle el departamento de Administración y Finanzas, del cual es parte el área de Soluciones y Proyectos, donde se realizó el diseño e implementación del formulario automatizado.

**Imagen 1: Organigrama de SOBOCE S.A.**



Fuente: Elaboración propia con base en la documentación de SOBOCE S.A.

### Área de Tecnología y Procesos

El área de Tecnología y Procesos es responsable de administrar y optimizar los sistemas de información, procesos y herramientas tecnológicas utilizadas en la empresa. Esta área se enfoca en el diseño, desarrollo, implementación y mantenimiento de sistemas y tecnologías que permitan mejorar la eficiencia y eficacia de los procesos empresariales. Además, es responsable de garantizar que los sistemas y tecnologías utilizados sean seguros y estén actualizados.

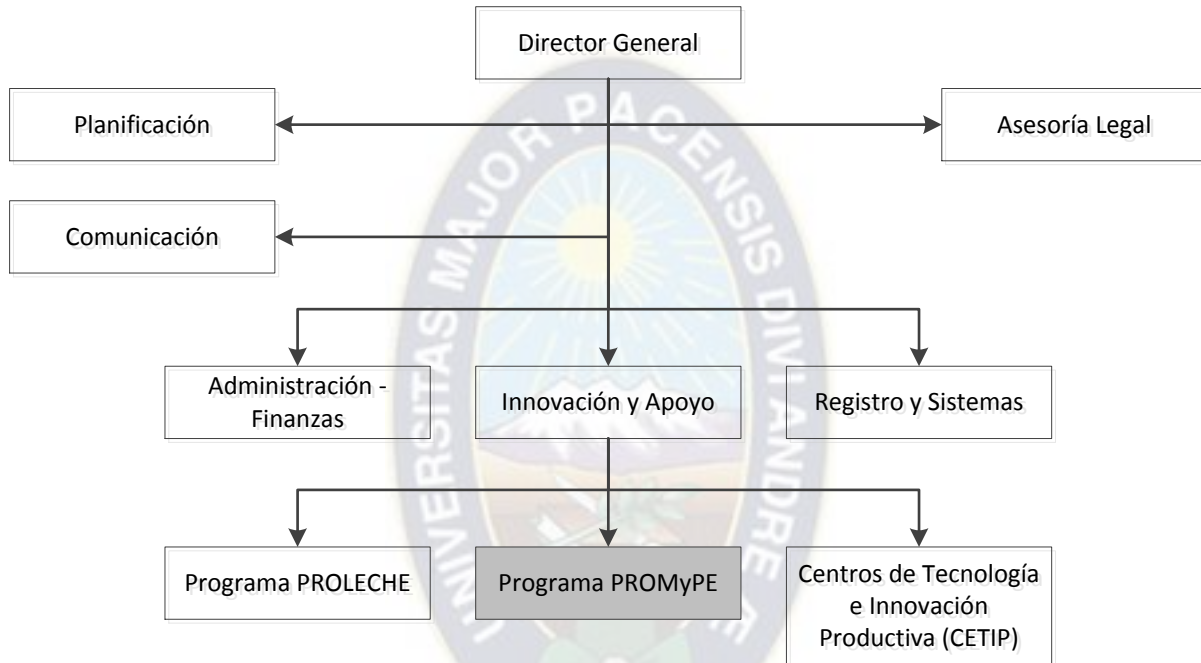
### PRO-BOLIVIA

PROBOLIVIA ejecuta el programa PROMyPE/Jiwasá desde el año 2014 con el objetivo principal de impulsar la matriz productiva de Bolivia mediante la modernización y tecnificación, generando valor agregado a la producción artesanal, agroindustrial, manufacturera e industrial. Este programa consiste en la elaboración de un plan de mejora que es un documento de evaluación técnica y financiera de unidades productiva para aprobar el cofinanciamiento de activos productivos (materia prima, insumos y maquinaria) y asistencia técnica. Este programa finalizó su primera fase el año 2020 con apoyo de la Embajada de Dinamarca (DANIDA) y la Cooperación Suiza en Bolivia (COSUDE).

## Organización

La estructura organizativa de PRO-BOLIVIA se detalla en la Imagen 2. La toma de decisiones está a cargo del director general, el área de innovación y apoyo esta encargada de los programas PROLECHE, PROMyPE y CETIP que cuenta con oficinas regionales en los departamentos de La Paz, Cochabamba, Oruro, Potosí, Chuquisaca y Tarija.

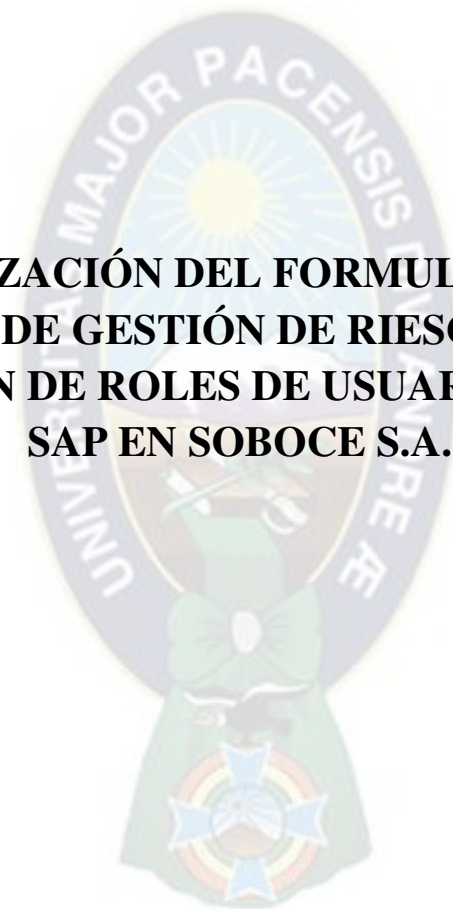
**Imagen 2: Organigrama de PRO-BOLIVIA**



Fuente: Elaboración propia con base en el sitio web de PRO-BOLIVIA <https://www.probolivia.gob.bo/organigrama/> (consultado el 25 de abril de 2023)

## Área II - Caso de estudio

**AUTOMATIZACIÓN DEL FORMULARIO PARA EL  
PROCEDIMIENTO DE GESTIÓN DE RIESGOS PARA EL ALTA  
O MODIFICACIÓN DE ROLES DE USUARIOS DEL SISTEMA  
SAP EN SOBOCE S.A.**



# Capítulo 1: Introducción

La Ingeniería Industrial se enfoca en optimizar procesos y sistemas empresariales para mejorar su rendimiento. En esta memoria laboral, se presenta un caso de estudio sobre la aplicación de los conocimientos adquiridos por el autor en la empresa SOBOCE S.A., destacando la importancia del desarrollo de herramientas personalizadas de software.

El caso de estudio se centra en el procedimiento denominado por SOBOCE S.A. como “GESTIÓN DE RIESGOS PARA EL ALTA O MODIFICACIÓN DE ROLES” que tiene la finalidad de generar un formulario con los datos necesarios para evaluar y gestionar los riesgos asociados a los roles de usuario en el sistema SAP y como el autor plantea una solución al problema de la ineficiencia en el llenado del formulario utilizado en este procedimiento.

El objetivo general de este estudio es realizar la automatización de dicho formulario, buscando optimizar el procedimiento, para ello el autor empleó el enfoque de procesos para llevar a cabo un análisis detallado del procedimiento con la finalidad de identificar las causas, los requisitos necesarios para diseñar y desarrollar una herramienta personalizada de software utilizando el editor de código de Microsoft Excel, esta herramienta se denominará "formulario automatizado" a lo largo de la presente memoria laboral.

## 1.1. Planteamiento del problema

### 1.1.1. Análisis causa-efecto

El diagrama causa-efecto desarrollado por Kaoru Ishikawa es un método que consiste en definir la ocurrencia de un problema no deseable (efecto), para después identificar los factores que contribuyen al problema (causas). Por lo general, las principales causas se subdividen en cinco categorías principales, mano de obra (talento humano), maquinaria o herramienta, métodos o procesos, medio ambiente, dirección. Cada una de estas categorías se subdividen en sub causas, el proceso continúa hasta que se detectan todas las causas posibles que contribuyen al problema. (Niebel & Freivalds, 2009)

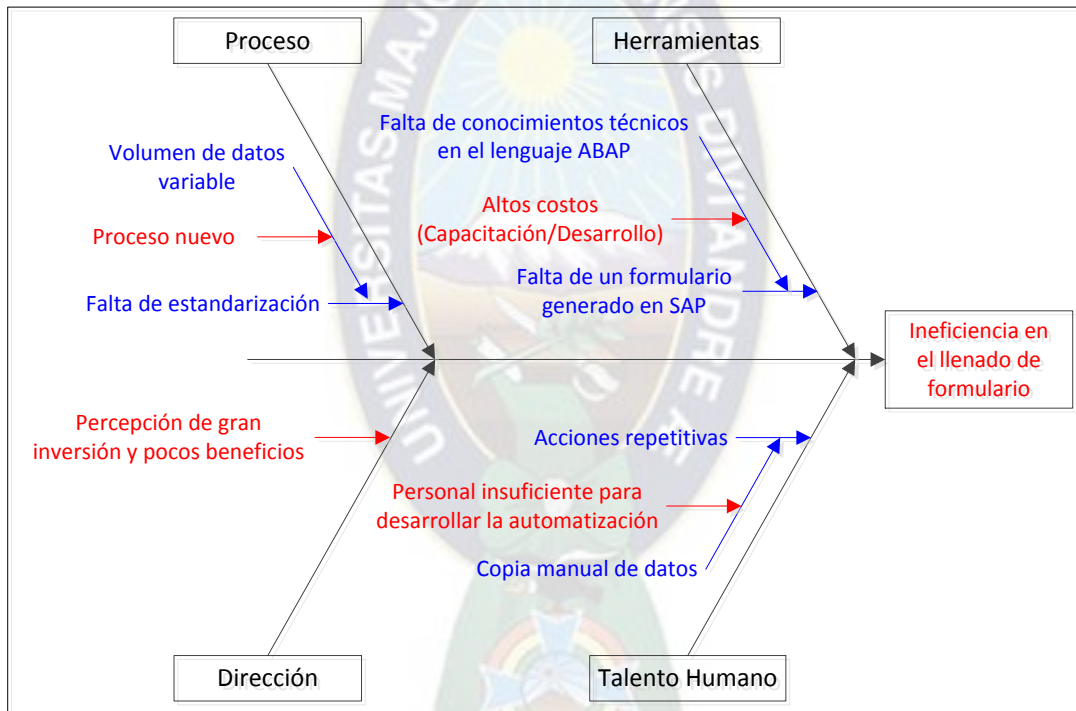
Mediante el diagrama de causa-efecto (ver Imagen 3), se identifican las posibles causas de la deficiencia en el desarrollo de herramientas personalizadas para la automatización de tareas. La Tabla 2 describe las categorías empleadas en el diagrama.

**Tabla 2: Categorías del diagrama de Ishikawa**

<b>Categoría</b>	<b>Descripción</b>
Herramientas	Software utilizado en el llenado del formulario.
Talento Humano	Personal encargado del llenado del formulario.
Dirección	Responsables en la toma de decisiones referentes al procedimiento “Gestión de riesgos para el alta o modificación de roles”.
Procesos	Inputs y Acciones necesarias para realizar el llenado del formulario.

Fuente: Elaboración propia con base en Niebel & Freivalds (2009)

**Imagen 3: Diagrama causa-efecto (diagrama de Ishikawa)**



Fuente: Elaboración propia.

En el contexto del procedimiento "Gestión de riesgos para el alta o modificación de roles", se han identificado las causas de la ineficiencia en el llenado del formulario utilizado en dicho procedimiento a través de un análisis de causa-efecto. Entre las causas identificadas se encuentran la novedad del proceso, la falta de personal con las habilidades y conocimientos necesarios en el lenguaje ABAP para desarrollar un reporte automático en el sistema SAP, así como los altos costos asociados a la capacitación o contratación de consultores para el desarrollo. Además, existe la percepción de que la inversión requerida supera los beneficios para la empresa. Estas limitaciones han generado una carga de trabajo adicional para el personal actual y han ocasionado retrasos en



el manejo de los riesgos existentes al utilizar una herramienta “manual” para gestionar los datos del procedimiento.

### **1.1.2. Problema**

Ineficiencia en el llenado del formulario para el procedimiento de gestión de riesgos para el alta o modificación de roles de usuarios del sistema SAP en SOBOCE S.A.

### **1.2. Objetivo general**

Realizar la automatización del formulario para el procedimiento de gestión de riesgos para el alta o modificación de roles de usuarios del sistema SAP en SOBOCE S.A.

### **1.3. Objetivos específicos**

- 1) Sistematizar la información secundaria relacionada con la automatización de Microsoft Excel.
- 2) Diagnosticar el procedimiento de gestión de riesgos para el alta o modificación de roles de usuarios del sistema SAP en SOBOCE S.A.
- 3) Desarrollar el software con base en los requerimientos del procedimiento de gestión de riesgos para el alta o modificación de roles de usuarios del sistema SAP en SOBOCE S.A.
- 4) Evaluar el impacto del software desarrollado.

### **1.4. Justificación**

#### **1.4.1. Justificación Económico-Social**

La creación e implementación de herramientas personalizadas de software con el objetivo de automatizar tareas administrativas rutinarias, permite reducir los errores humanos y se optimiza el uso de recursos, lo que contribuye a un uso más eficiente de los recursos empresariales y ahorra costos en términos de tiempo y mano de obra. Estas herramientas personalizadas de software también representan una alternativa económica frente a las soluciones comerciales disponibles en el mercado.

En el aspecto social, la implementación de herramientas personalizadas también beneficia al personal involucrado al reducir la carga de trabajo. Además, al adquirir habilidades técnicas relacionadas con el desarrollo de software, el personal mejora su empleabilidad y su capacidad para adaptarse a los cambios en el mercado laboral.

En un enfoque integral, el fortalecimiento de las empresas mediante la implementación de nuevas tecnologías, el fomento de la creatividad y la búsqueda de soluciones por parte de su personal contribuye en ciertos aspectos como a la estabilidad económica, la generación de empleo, la innovación, lo que a su vez tiene un impacto positivo en la sociedad al impulsar el desarrollo económico y mejorar el nivel de vida de la sociedad en general.

#### **1.4.2. Justificación académica**

Abordar el tema de la automatización de un formulario es importante porque contribuye al conocimiento existente, resuelve un problema específico en la empresa SOBOCE S.A. con base

en los conocimientos adquiridos por el autor y su aplicación en el campo laboral (ver Tabla 3). Además, la presente memoria laboral aportar enfoques técnicos, metodológicos y buenas prácticas que pueden ser aplicables en otras organizaciones y sistemas similares.

**Tabla 3: Conocimientos adquiridos y aplicados**

<b>Materia</b>	<b>Conocimiento adquirido</b>	<b>Aplicación</b>
Informática para ingeniería	Sistemas operativos, razonamiento lógico, algoritmos y programación estructurada para la solución de problemas.	Desarrollo de herramientas complementarias para las aplicaciones de Microsoft Office.
Ingeniería de métodos	Análisis y optimización de procesos de producción y/o servicios.	Relevamiento de datos para el análisis y planteamiento de mejoras en un proceso.
Ingeniería de sistemas	Conceptos y técnicas metodológicas de sistemas para la identificación de problemas y planteamiento de alternativas de solución enmarcadas en los diferentes enfoques del pensamiento sistémico.	Planteamiento de soluciones derivadas del análisis de procesos: <ul style="list-style-type: none"> <li>• Identificación de componentes y su interacción</li> <li>• Identificación de problemas y sus efectos.</li> </ul>
Automatización industrial	Esquemmatización de soluciones complejas sobre la base de la reducción de tiempos improductivos, obtención de información de variables para el control de procesos y soluciones integrales ERP.	Propuesta, diseño y desarrollo de herramientas para la automatización de tareas relacionadas con el manejo de datos de diferentes sistemas.
Logística	Conceptos de almacenamiento de productos, control de inventarios, y distribución.	Gestión de los datos relacionados con la recepción de materia prima y control de inventarios para la elaboración de reportes para la toma de decisiones en planta.

Fuente: Elaboración propia con base en el plan de estudios (2015) de la carrera de Ingeniería Industrial de la Universidad Mayor de San Andrés.



## Capítulo 2: Marco Teórico

En la actualidad, la automatización de procesos y el desarrollo de software desempeñan un papel fundamental en el mundo empresarial. Con el avance de la tecnología de la información, se ha vuelto esencial contar con sistemas eficientes y herramientas adecuadas para optimizar y agilizar los procesos organizacionales. En este marco teórico, se presentan conceptos relacionados con la automatización, el desarrollo de software y los sistemas de información, con el objetivo de comprender su impacto en los procesos de negocio.

Asimismo, se detallan herramientas específicas como Microsoft Excel, VBA (Visual Basic for Applications) y SQL (Structured Query Language), que son ampliamente utilizados en el desarrollo de software y en el análisis de datos. Por último, se abordará el impacto de los sistemas de información y el desarrollo de software personalizado en la organización y en los procesos de negocio.

### 2.1. Automatización

Automatización es un término que se utiliza para aplicaciones de tecnología en las que se minimiza la interacción humana. La automatización básica consiste en digitalizar el trabajo usando herramientas para optimizar las tareas rutinarias. La automatización de procesos logra uniformidad y eficiencia mediante aplicaciones empresariales y software dedicado. En los últimos años, la automatización de integración ha logrado la imitación de las tareas de un ser humano mediante software entrenado (robots). El tipo de automatización más reciente emplea inteligencia artificial donde el software toma decisiones basándose en el aprendizaje de experiencias previas. (IBM, 2023)

En el contexto actual, las instituciones y organizaciones necesitan implementar sistemas de gestión documental y automatización de procesos administrativos. Los sistemas de gestión documental permiten la digitalización y administración de documentos, mientras que la automatización de procesos agiliza las tareas administrativas y optimiza los flujos de trabajo. Estos sistemas contribuyen a mejorar la gestión administrativa gubernamental y corporativa, diseñar y auditar flujos de trabajo, y mantener sistemas administrativos certificados según la norma ISO (Franklin Fincowsky, 2009).

Considerado lo anterior y sobre la base de la experiencia del autor (ver Tabla 1, pág. 2), se puede definir la automatización como la implementación de herramientas y soluciones tecnológicas que reduzcan la carga de trabajo en actividades administrativas rutinarias, mejorando así la eficiencia y precisión en su cumplimiento. Entonces, de forma general se puede afirmar que la aplicación práctica de la automatización en áreas administrativas tiene

como objetivo optimizar los procesos, reducir errores y mejorar la eficiencia en la gestión de tareas.

## 2.2. Software

Según Pressman (2010) el software es un elemento de un sistema lógico que brinda el potencial para transformar datos, administrar la información de negocios, proveer una red mundial de información (internet) y brindar los medios para obtener información en todas sus formas. Pressman también menciona la existencia de siete categorías principales de software, las cuales se detallan en la Tabla 4.

**Tabla 4: Categorías de Software**

<b>Categoría</b>	<b>Descripción</b>
Software de sistemas	Conjunto de programas que brindan servicios a otros programas (Windows, Linux, macOS).
Software de aplicación	Programas aislados que resuelven necesidades específicas de negocios (Microsoft Office, Adobe Photoshop, etc.).
Software de ingeniería y ciencias	Utilizado en aplicaciones técnicas y científicas, desde astronomía hasta manufactura automatizada (MATLAB, AutoCAD, RStudio).
Software incrustado	Reside dentro de productos o sistemas para implementar y controlar características y funciones para el usuario final (Control numérico computarizado, Firmware en dispositivos de red, etc.).
Software de línea de productos	Diseñado para proporcionar capacidades específicas para múltiples consumidores (Salesforce, CRM, SAP, etc.).
Aplicaciones web	Software centrado en redes que abarca una amplia gama de aplicaciones en línea (Facebook, Google Drive, Amazon, etc.).
Software de inteligencia artificial	Utiliza algoritmos no numéricos para resolver problemas complejos que no son fáciles de tratar computacionalmente (Siri, ChatGPT, etc.).

Fuente: Elaboración propia con base en Pressman (2010)

### 2.2.1. Desarrollo de software

Según Pressman (2010) el proceso de desarrollo de software establece actividades estructurales que son aplicables a todos los proyectos, sin importar su tamaño o complejidad. Estas actividades incluyen la comunicación, la planificación, el modelado, la construcción y el despliegue. La Tabla 5 proporciona detalles sobre estas actividades en el contexto del desarrollo de software.

**Tabla 5: Actividades estructurales del desarrollo de software**

<b>Actividad</b>	<b>Descripción</b>
Comunicación	Reuniones con el cliente, investigación e identificación de requerimientos.
Planeación	Creación de un plan que define las tareas técnicas, riesgos, recursos, productos del trabajo y programación de actividades.
Modelado	Creación de representaciones visuales y abstractas que permiten comprender, validar los requerimientos y definir el diseño antes de pasar a la etapa de construcción.
Construcción	Generación de código y realización de pruebas técnicas para descubrir errores.
Despliegue	Implementación del software para su evaluación y retroalimentación por parte de los usuarios. Además, se realiza el mantenimiento continuo para asegurar el funcionamiento y la adaptabilidad del software.

Fuente: Elaboración propia con base en Pressman (2010)

El modelado desempeña un papel fundamental en el desarrollo de software, ya que permite crear representaciones abstractas de sistemas con distintos niveles de precisión y detalle. Antes de implementar un sistema, se construyen modelos que proporcionan una mejor comprensión del sistema en desarrollo, estableciendo un conjunto de elementos y reglas que guían el desarrollo (Gomaa, 2011).

### **2.2.2. Modelo**

Un modelo es un enfoque de proceso que establece un conjunto de elementos y reglas para guiar el desarrollo de software. Cada modelo proporciona instrucciones detalladas sobre las actividades estructurales, acciones de ingeniería de software, tareas, productos, aseguramiento de la calidad y mecanismos de control. Además, cada modelo define un flujo de trabajo que establece cómo se relacionan entre sí los diferentes elementos del proceso. Estos modelos buscan aportar estructura y organización al trabajo de desarrollo de software, aunque plantean interrogantes sobre su idoneidad en un entorno que requiere adaptabilidad y flexibilidad (Pressman, 2010). La Tabla 6 detalla los principales modelos para el desarrollo de software.

**Tabla 6: Modelos para el desarrollo de software**

<b>Modelo</b>	<b>Descripción</b>
Cascada	Enfoque sistemático y secuencial adecuado para adaptaciones o mejoras definidas en sistemas existentes y proyectos con requerimientos claros y estables. El proceso comienza con la especificación de los requerimientos por parte del cliente y avanza a través de etapas como planeación, modelado, construcción y despliegue, hasta llegar al soporte del software terminado.
V	Enfoque que muestra la relación entre las actividades de desarrollo y las actividades de verificación y validación. El lado izquierdo de la "V" representa la fase de desarrollo (requisitos, diseño, codificación e integración). El lado derecho de la "V" representa la fase de verificación y validación. El modelo en V ayuda a garantizar que el software se construya de acuerdo con los requisitos establecidos y que se realicen las pruebas necesarias para validar su funcionamiento.
Incremental	Enfoque basado en la entrega gradual y secuencial de incrementos funcionales del producto final. Cada incremento consiste en una versión operativa del software que cumple con los requisitos básicos y proporciona funcionalidad al usuario. Este modelo permite una implementación progresiva del proyecto, para gestionar riesgos técnicos y evitar retrasos importantes.
Prototipos	Enfoque utilizado cuando los requerimientos no están claros o necesitan una mayor definición. Consiste en la creación de prototipos rápidos que representan aspectos visibles del software, como la interfaz de usuario o los formatos de salida. Estos prototipos se entregan y se evalúan, lo que permite obtener retroalimentación para mejorar los requerimientos.
Espiral	Enfoque cíclico y evolutivo que combina elementos de la iteratividad de hacer prototipos con la estructura controlada del modelo en cascada. Se desarrolla en entregas evolutivas, comenzando con la especificación del producto y continuando con el desarrollo de prototipos y versiones cada vez más sofisticadas del software.
Concurrente	Enfoque que permite realizar actividades de otros modelos de manera simultánea en lugar de seguir una secuencia lineal. Es aplicable a todos los tipos de desarrollo de software y proporciona una visión clara del estado actual del proyecto al establecer una red del proceso en la cual las actividades existen simultáneamente y los eventos generados impulsan las transiciones.

(Continúa en la siguiente página)

(Continuación)

<b>Modelo</b>	<b>Descripción</b>
Componentes	Enfoque evolutivo e iterativo centrado en la reutilización de fragmentos de software comercial prefabricado conocidos como componentes para acelerar y optimizar el proceso de desarrollo de aplicaciones. El modelo se basa en identificar y evaluar los componentes disponibles, se diseña una arquitectura para recibirlos, se integran en la estructura del software y se realizan pruebas exhaustivas para garantizar su funcionalidad.
Métodos formales	Enfoque centrado en la especificación matemática rigurosa del software. Permite eliminar problemas de ambigüedad, incompletitud e inconsistencia mediante el análisis matemático, lo que facilita la detección y corrección de errores. Este modelo requiere tiempo, esfuerzo y capacitación especializada y se utiliza en áreas donde la calidad y seguridad del software son críticas, como el control electrónico de aeronaves y equipos médicos.
Orientado a aspectos	Enfoque centrado en abordar aspectos globales que afectan múltiples partes de un sistema. Estos aspectos se definen como propiedades que tienen un efecto a través de la arquitectura del software. El modelo implica descomponer el software en componentes pequeños y especializados (por ejemplo manejo de pedidos, inventario y pagos) y agrupar estos componentes en conjuntos que se ocupen de preocupaciones globales, como la seguridad del sistema o la experiencia del usuario.

Fuente: Elaboración propia con base en Pressman (2010)

Un modelo de desarrollo proporciona la metodología general para el desarrollo del software, mientras que la arquitectura de software se centra en la estructura y diseño específicos del sistema.

### **2.2.2.1. Prototipo**

Un prototipo es una representación temprana o parcial de un producto, sistema o proyecto que brinda la oportunidad de obtener retroalimentación de los usuarios y realizar ajustes o mejoras antes de comprometer recursos adicionales en el desarrollo. La recopilación de información durante la fase de prototipado permite establecer prioridades y realizar modificaciones sin sufrir grandes consecuencias o interrupciones significativas. Debido a esta característica, la creación de prototipos y la planificación están estrechamente vinculadas. (Kendall & Kendall, 2011)

### **2.2.3. Arquitectura de software**

La arquitectura de software se refiere al conjunto de estructuras necesarias para comprender un sistema que involucra componentes de software (módulos, librerías, servicios, clases, base de datos, entre otros), sus relaciones y las propiedades tanto del sistema como de los



componentes (Cervantes & Kazman, 2016). Las propiedades se refieren a las características específicas de cada componente, la relación entre los componentes se centra en cómo se comunican e interactúan entre sí para lograr los objetivos del sistema (Pressman, 2010).

La arquitectura de software es una representación que permite analizar el diseño, considerar alternativas arquitectónicas y reducir riesgos durante la construcción del software. Existe una diferencia entre la arquitectura y el diseño de software, siendo este último una instancia específica de una arquitectura (Pressman, 2010). Es decir, que la arquitectura de software se enfoca en la estructura global y las decisiones clave, mientras que el diseño de software se ocupa de los detalles de implementación a nivel de componentes individuales.

Un estilo de arquitectura establece una estructura para todos los componentes del sistema y puede implicar cambios fundamentales en la estructura del software. A diferencia de un estilo, un patrón de arquitectura se enfoca en aspectos específicos y establece reglas para el manejo de la funcionalidad del software. Los patrones se utilizan junto con un estilo para dar forma a la estructura general del sistema de software (Pressman, 2010). La Tabla 7 detalla los principales estilos de arquitectura de software.

**Tabla 7: Estilos de arquitectura de software**

<b>Estilo</b>	<b>Descripción</b>
Centrado en los datos	Se basa en tener un lugar central donde se guardan los datos (archivo o una base de datos) y otros componentes pueden acceder a estos datos para hacer cambios, agregar nueva información o eliminar datos. Estas arquitecturas permiten que diferentes partes del software trabajen de forma independiente, sin depender de los cambios que hagan otros usuarios.
Flujo de datos	Se utiliza cuando se necesita transformar datos de entrada en datos de salida a través de diferentes componentes. Un patrón común en este tipo de arquitectura es el de tubo y filtro, donde los componentes llamados filtros están conectados por tubos que transmiten los datos de un filtro al siguiente. Cada filtro funciona de manera independiente y espera una entrada de datos en un formato específico, produciendo datos de salida para el siguiente filtro.
Llamado y retorno	Se basa en la subdivisión de la funcionalidad del sistema en componentes más pequeños y reutilizables. El componente principal invoca a otros componentes para realizar tareas específicas y luego regresa al componente principal para continuar su ejecución. Esto permite una estructura de programa modular y escalable, facilitando la modificación y mantenimiento del sistema.

(Continúa en la siguiente página)

(Continuación)

<b>Estilo</b>	<b>Descripción</b>
Orientado a objetos	Enfoque donde los componentes del sistema son objetos que contienen tanto datos como las operaciones necesarias para manipular esos datos. En este estilo la comunicación y coordinación entre los componentes se realiza mediante el envío de mensajes. Esto permite la interacción y colaboración entre los objetos para lograr los objetivos del sistema.
En capas	Enfoque en el que un sistema se divide en capas separadas y distintas. Cada capa tiene una función específica y se construye sobre la capa anterior. Las capas internas se ocupan de la interacción con el sistema operativo, las capas intermedias proporcionan servicios y funciones adicionales y las capas externas están relacionadas con la interfaz de usuario. Este enfoque permite una separación clara de responsabilidades, además de facilitar la modularidad y escalabilidad del sistema.

Fuente: Elaboración propia con base en Pressman (2010)

#### **2.2.4. Lenguaje de Programación**

Un lenguaje de programación es un conjunto de instrucciones, símbolos, reglas de sintaxis y semántica que permiten a los programadores escribir código fuente, el cual se compila a código máquina para que un ordenador u otro dispositivo pueda ejecutar el programa. Debido a las diferentes necesidades tecnológicas, se han creado varios tipos de lenguajes de programación, como Python, C++, C#, Java, JavaScript, VB, VBA, entre otros. Cada lenguaje tiene sus propias características, sintaxis y reglas que determinan cómo se debe escribir el código correspondiente (Joyanes Aguilar, 2008).

Con base en la experiencia del autor, aprender un lenguaje de programación contribuye al desarrollo de habilidades lógicas y analíticas, la estimulación de la creatividad, incremento de la eficiencia mediante la automatización de tareas y la resolución de problemas. Además, adquirir estas habilidades amplía las oportunidades profesionales.

#### **2.2.5. Algoritmo**

Donal Knuth (1997) un referente sobre algoritmos y estructura de datos, define a un algoritmo como un conjunto de instrucciones o reglas bien definidas y ordenadas que permiten realizar una tarea en un determinado número de pasos. Establece También que un algoritmo tiene cinco características importantes, finitud (siempre termina después de un número determinado de pasos), definición (especificación rigurosa y unívoca de cada acción a través de un lenguaje de programación), entradas (valores entregados antes o durante la ejecución del algoritmo), salidas (valores obtenidos por la ejecución del algoritmo y relacionados con las entradas.) y eficacia (Cada paso debe ser ejecutable).

De la misma forma, Cormen *et al.* (2022) definen un algoritmo como una herramienta para resolver un problema mediante una secuencia de instrucciones que transforman las entradas en salidas (solución). Entonces el algoritmo es correcto si, para cada entrada proporcionada, finaliza el cálculo o procesamiento en un tiempo finito y genera la solución correcta para la instancia del problema. La especificación del algoritmo puede realizarse de forma escrita, mediante un lenguaje de programación o incluso diagramas de diseño. El único requisito es que la especificación proporcione una descripción precisa del procedimiento a seguir.

Un algoritmo es independiente del lenguaje de programación y sirve como infraestructura para resolver problemas. Los algoritmos se clasifican en cualitativos (no involucran cálculos numéricos) y cuantitativos (involucran cálculos numéricos). Un programa, por otro lado, es un conjunto de instrucciones codificadas en un lenguaje de programación que expresa uno o varios algoritmos y puede ser ejecutado en una computadora (Bel, 2020).

### 2.2.6. Ingeniería de requerimientos

La comprensión de los requerimientos en el desarrollo de software es fundamental para el éxito del proyecto. Aunque algunos argumentan que es suficiente empezar a elaborar el software y ajustar los requerimientos sobre la marcha, esta mentalidad puede llevar al fracaso del proyecto. La ingeniería de requerimientos abarca una serie de tareas y técnicas que permiten comprender y especificar las necesidades del negocio, los escenarios de uso, las funciones, las características y las restricciones del proyecto. Esta disciplina sirve como puente entre el diseño y la construcción del software, permitiendo analizar el contexto del trabajo, establecer prioridades y definir una solución clara y sin ambigüedades (Pressman, 2010).

La ingeniería de requerimientos incluye tareas como la concepción, la indagación, la elaboración, la negociación, la especificación, la validación y la administración de los requerimientos. La Tabla 8 detalla las tareas de la ingeniería de requerimientos. Es importante destacar que estas tareas pueden ocurrir en paralelo y se adaptan a las necesidades específicas de cada proyecto.

**Tabla 8: Tareas de la ingeniería de requerimientos**

<b>Tarea</b>	<b>Descripción</b>
Concepción	Se establece el entendimiento básico del problema, las personas involucradas, la naturaleza de la solución y se establece una comunicación preliminar entre los participantes y el equipo de software.
Indagación	Se recopila información sobre los objetivos del sistema o producto. Sin embargo, es una tarea difícil, ya que los clientes y usuarios pueden tener dificultades para expresar claramente sus necesidades.

(continúa en la siguiente página)



(continuación)

<b>Tarea</b>	<b>Descripción</b>
Elaboración	La información obtenida en la concepción e indagación se convierte en un modelo detallado de requerimientos que incluye aspectos funcionales, de comportamiento e información del software.
Negociación	Implica reconciliar conflictos y priorizar los requerimientos. Las solicitudes pueden ser mayores a lo que se puede lograr, por lo que se lleva a cabo un proceso de negociación donde se realizan ajustes para satisfacer a todas las partes involucradas.
Especificación	Presentación clara y comprensible de los requerimientos mediante documentos escritos, modelos gráficos, modelos matemáticos, escenarios de uso o prototipos.
Validación	Se analiza la especificación para garantizar que los requerimientos estén claros, sin ambigüedades, y se identifican y corrigen inconsistencias, omisiones y errores. La revisión técnica es un mecanismo importante para validar los requerimientos.
Administración de los requerimientos	Implica identificar, controlar y dar seguimiento a los cambios en los requerimientos en cualquier etapa del proyecto.

Fuente: Elaboración propia con base en Pressman (2010)

### **2.2.7. Calidad del software**

Pressman (2010) define la calidad del software como el proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan. Esta definición enfatiza tres aspectos importantes: un proceso eficaz establece la infraestructura de apoyo a la elaboración del software; un producto útil entrega contenido, funciones y características que el usuario final desea; agregar valor para beneficiar a los productores y a los usuarios finales.

En el contexto del desarrollo de software, los estándares de calidad surgieron en la década de 1970, inicialmente en contratos para software militar y luego en el desarrollo de software comercial. El aseguramiento de la calidad del software se refiere a un conjunto planificado y sistemático de acciones para garantizar una alta calidad en el software. Es responsabilidad de varias entidades, incluyendo ingenieros de software, gerentes de proyecto, clientes y el grupo de Aseguramiento y Control de la Calidad del software (ACS) (Pressman, 2010).

La Tabla 9 detalla los elementos clave que garantizan la calidad en el desarrollo y entrega de productos.

**Tabla 9: Elementos de la calidad de software**

<b>Elemento</b>	<b>Descripción</b>
Estándares	Se refiere a los estándares establecidos por organizaciones como el IEEE, ISO entre otras.
Revisiones	Las revisiones técnicas son realizadas por ingenieros de software para detectar errores. Las auditorías
Auditorías	Son llevadas a cabo por el personal del ACS para garantizar el cumplimiento de los lineamientos de calidad.
Pruebas	Tienen como objetivo principal detectar errores en el software.
Administración del cambio	Implementación de prácticas adecuadas de administración de cambios para evitar confusiones y mantener la calidad.
Educación	Capacitación y promoción de la educación de los ingenieros de software, gerentes y otros participantes.
Administración de proveedores	Establecimiento de prácticas y cláusulas de calidad en los contratos con proveedores externos.
Administración de la seguridad	Establecimiento de políticas y utilizar procesos y tecnologías adecuadas para garantizar la seguridad del software.
Seguridad	Dado que el software puede ser un componente crucial en sistemas críticos, se evalúa el efecto de las fallas del software y se toman las medidas necesarias para reducir el riesgo.
Administración de riesgos	Aseguramiento de la realización adecuada de las actividades de administración de riesgos y se establecimiento de planes de contingencia relacionados con los riesgos.

Fuente: Elaboración propia con base en Pressman (2010)

### **2.3. Sistema**

La definición más citada y mayormente aceptada sobre un sistema es la de Ludwig Von Bertalanffy (1968), fundador de la teoría general de los sistemas, quien definió un sistema como un conjunto de elementos interconectados y relacionados, que forman un todo. En otras palabras, un sistema es un conjunto de componentes que interactúan entre sí y con su entorno para cumplir un propósito o una función determinada.

Wasson (2006) de la misma forma que de Ludwig Von Bertalanffy define un sistema como un conjunto integrado de elementos interdependientes, donde cada uno de estos elementos tiene capacidades delimitadas y específicas, que funcionan de forma conjunta para permitir la satisfacción de necesidades operativas orientadas a un resultado específico y con probabilidad de éxito.

En el contexto de una organización, un sistema se refiere a una red de canales de comunicación, esta red permite la interacción y el flujo de información entre los diferentes procesos y partes interesadas relevantes. Los procesos dentro del sistema están interconectados, ya que las salidas de un proceso pueden convertirse en las entradas de otro proceso (International Organization for Standardization [ISO], 2015).

### **2.3.1. Proceso**

Un proceso es un conjunto de actividades mutuamente relacionadas que utilizan las entradas en un resultado previsto. Se logran resultados coherentes y predecibles de manera más eficiente y efectiva cuando las actividades se comprenden y gestionan como procesos interrelacionados que operan como un sistema coherente (International Organization for Standardization [ISO], 2015).

En el contexto de la ingeniería de software, un proceso se refiere a un conjunto de actividades, acciones y tareas que se llevan a cabo para crear un producto. Las actividades tienen objetivos amplios, las acciones son conjuntos de tareas que producen resultados importantes, y las tareas se enfocan en objetivos específicos. El proceso no es una prescripción rígida, sino un enfoque adaptable que permite elegir las acciones y tareas adecuadas. El objetivo es entregar software de calidad en tiempo oportuno para satisfacer a los patrocinadores y usuarios (Pressman, 2010).

### **2.3.2. Enfoque en procesos**

El enfoque a procesos implica la definición y gestión sistemática de los procesos y sus interacciones para alcanzar los resultados previstos. Permite comprender y cumplir los requisitos, agregar valor, lograr un desempeño eficaz y mejorar continuamente a través de la evaluación de datos e información (International Organization for Standardization [ISO], 2015)

Según la experiencia del autor al aplicar el enfoque a procesos en el desarrollo de software, se pueden obtener los siguientes beneficios:

- 1) **Comprensión de los requisitos**, mediante el establecimiento de procesos para analizar, documentar y validar los requisitos de forma clara y precisa, lo que facilita su cumplimiento y evita malentendidos.
- 2) **Desempeño eficaz**, mediante la planificación, ejecución y control, lo que facilita la toma de decisiones para el desarrollo de software.
- 3) **Mejora continua**, mediante la identificación de oportunidades de mejora, la eliminación de actividades innecesarias, la reducción de errores y mejorar el software en cada iteración.

### **2.3.3. Procedimiento**

Un procedimiento es la forma específica de llevar a cabo una tarea o actividad (International Organization for Standardization [ISO], 2015). En concordancia con esto, en SOBOCE S.A. (2018) se define un procedimiento como un documento donde se describe de forma específica un proceso completo o los subprocesos principales que lo componen en función a la complejidad de estos, con el objetivo de que el proceso alcance de manera eficaz los resultados previstos.

### **2.3.4. Sistema de información**

Un sistema de información es un conjunto de componentes interrelacionados que se encarga de recolectar, procesar, almacenar y distribuir información para apoyar en la toma de decisiones y el control en una organización. El sistema de información también ayuda a analizar problemas, visualizar temas complejos y crear nuevos productos. Aunque los sistemas de información están basados en la tecnología informática, hay una clara distinción entre una computadora, un programa informático y un sistema de información. Es decir que las computadoras y los programas informáticos son herramientas de los sistemas de información, pero no son el sistema en sí. Por lo tanto, es fundamental conocer el funcionamiento de las computadoras y los programas informáticos de un sistema de información para apoyar en la solución de problemas organizacionales (Laudon & Laudon, 2012).

El desarrollo de un sistema utiliza la información recopilada para diseñar de manera lógica el sistema de información. Esto implica diseñar procedimientos para garantizar la entrada precisa de datos al sistema y el diseño de una interfaz de usuario atractiva. Además, se realiza el diseño de la base de datos para almacenar los datos necesarios y se diseñan salidas que satisfagan las necesidades de información de los usuarios. También se incluye el diseño de controles y procedimientos de respaldo para proteger el sistema y los datos, así como la creación de especificaciones para los programadores (Kendall & Kendall, 2011).

#### **2.3.4.1. Componentes de un sistema de información**

Los componentes de un sistema de información son los elementos fundamentales que forman parte de su estructura y funcionamiento. Estos componentes trabajan en conjunto para llevar a cabo actividades de entrada, procesamiento, salida, almacenamiento y control, con el objetivo de convertir los recursos de datos en productos de información útiles para los usuarios del sistema (O'Brien & Marakas, 2006). La Tabla 10 detalla las características de estos componentes.

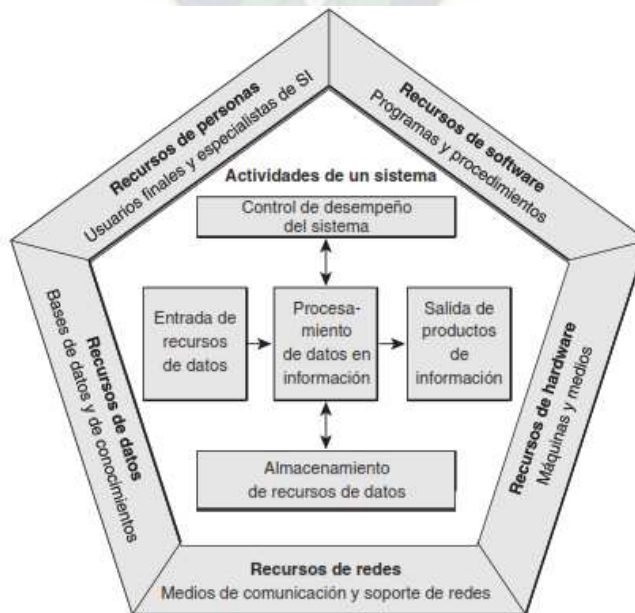
**Tabla 10: Componentes de un sistema de información**

<b>Componente</b>	<b>Descripción</b>
Personas	Involucra a los usuarios finales y especialistas en sistemas de información que utilizan y administran el sistema.
Hardware	Comprende las máquinas y los dispositivos físicos utilizados para procesar y almacenar datos, como computadoras, servidores, dispositivos de almacenamiento, impresoras, entre otros.
Software	Consiste en los programas y procedimientos utilizados para realizar diversas tareas en el sistema, como sistemas operativos, aplicaciones de software, software de gestión de bases de datos, entre otros.
Datos	Involucra las bases de datos y bases de conocimiento almacenado que contienen la información utilizada por el sistema. Se utilizan tecnologías de administración de datos para organizar, almacenar y obtener estos recursos.
Redes	Se refiere a los medios de comunicación y soporte de redes como internet o una intranet, entre otros, que permiten la transferencia de datos entre los componentes del sistema.

Fuente: Elaboración propia con base en O'brien & Marakas (2006)

La Imagen 4 representa un esquema conceptual fundamental de los componentes como recursos y las actividades principales de los sistemas de información.

**Imagen 4: Esquema de un sistema de información**



Fuente: O'brien & Marakas (Sistemas de información gerencial, 2006, pág. 26)



### 2.3.4.2. Tipos de sistemas de información

Los sistemas de información se clasifican en diferentes tipos, dependiendo de las necesidades de los usuarios y la organización. Algunos de estos tipos incluyen los sistemas de procesamiento de transacciones, sistemas de automatización de oficinas, sistemas de trabajo de conocimiento, sistemas de información administrativa, sistemas de soporte de decisiones y sistemas expertos. Cada sistema de información tiene su propósito y función específicos, contribuyendo a mejorar la eficiencia y efectividad de las operaciones y la toma de decisiones en las organizaciones. La Tabla 11 proporciona una visión general de los tipos de sistemas de información más relevantes.

**Tabla 11: Tipos de sistemas de información**

<b>Tipo</b>	<b>Característica</b>
Sistemas de Procesamiento de Transacciones	Procesan grandes cantidades de información para transacciones de negocios rutinarias
Sistemas de Automatización de Oficinas	Brindan soporte para analizar y transformar datos. Implica la generación y utilización de conocimiento dentro de la empresa.
Sistemas de Trabajo de Conocimiento	Brindan soporte a profesionales facilitando la creación, el análisis y la integración del conocimiento para resolver problemas complejos.
Sistemas de Información Administrativa	Brindan soporte para un amplio rango de tareas organizacionales, incluyendo análisis y toma de decisiones.
Sistemas de Soporte de Decisiones	Enfocados en brindar respaldo a la toma de decisiones en todas sus fases. Se adaptan a las necesidades de los usuarios y se consideran sistemas orientados hacia la inteligencia de negocios.
Sistemas Expertos	Utilizan la inteligencia artificial y la experiencia de expertos humanos para resolver problemas específicos.

Fuente: Elaboración propia con base en Kendall, K. E. & Kendall, J. E. (2011)

### 2.3.4.3. Organización

Una organización es una estructura social formal que utiliza los recursos del entorno para producir salidas. Se diferencia de un grupo informal por su estabilidad y duración en el tiempo, así como por su cumplimiento de leyes y reglas internas. Las organizaciones son entidades legales que establecen derechos, privilegios, obligaciones y responsabilidades. Además, se

caracterizan por desarrollar patrones de comportamiento, relaciones y acuerdos informales entre los miembros para llevar a cabo el trabajo (Laudon & Laudon, 2012).

Una organización empresarial se puede entender como un sistema integrado de información, conocido como sistema de planificación de recursos empresariales (ERP). Aunque su implementación puede ser desafiante, este sistema facilita el flujo de información entre las distintas áreas funcionales de la organización (Kendall & Kendall, 2011).

Los sistemas de información tienen un impacto en las políticas organizacionales al influir en el acceso a la información. Pueden afectar la asignación de tareas, la estructura y los procesos de una organización. La introducción de nuevos sistemas de información puede enfrentar resistencia debido a los cambios en las rutinas y la necesidad de capacitar a los empleados. La resistencia organizacional es un obstáculo común en la implementación de proyectos de tecnología de la información (Laudon & Laudon, 2012).

#### **2.3.4.4. Tecnología de la información**

La tecnología de la información (TI) se refiere al conjunto de recursos, herramientas y procesos utilizados para adquirir, almacenar, procesar, transmitir y utilizar la información de manera electrónica.

La integración de la tecnología en los sistemas implica combinar sistemas tradicionales con nuevas tecnologías para asegurar un contexto útil. Algunas de las tecnologías utilizadas incluyen aplicaciones de comercio electrónico y sistemas web, sistemas empresariales como los ERP, sistemas para dispositivos móviles, software de código abierto, entre otros. Sin embargo, la implementación de estas tecnologías puede presentar desafíos, como la adaptación de los sistemas existentes, cambios en la organización y capacitación especializada. El uso de tecnología integrada puede ser una estrategia efectiva para mejorar las operaciones y colaboración en las organizaciones (Kendall & Kendall, 2011).

La tecnología de la información afecta los costos y la economía de las empresas a medida que disminuye su costo, sustituyendo la mano de obra y otros capitales costosos. Esto permite a las empresas reducir su tamaño al externalizar tareas a proveedores externos en lugar de contratar empleados. Además, la tecnología de la información reduce los costos administrativos internos al facilitar la adquisición y análisis de información, lo que a su vez reduce los costos de agencia o coordinación (Laudon & Laudon, 2012).

#### **2.3.4.5. Sistemas de información en los procesos de negocio**

Los sistemas de información han transformado la forma en que se realizan los negocios. El uso de dispositivos móviles y el acceso a internet se han vuelto herramientas esenciales para las empresas. Cada vez más personas realizan compras en línea, investigan productos y servicios en internet. Las cadenas de suministro también se han agilizado, permitiendo una respuesta más rápida a la demanda de los clientes y una mayor eficiencia operativa (Laudon & Laudon, 2012).

Un proceso de negocio se refiere al conjunto de tareas y comportamientos que una organización lleva a cabo de manera lógica para alcanzar resultados específicos. En una empresa digital, estos procesos se realizan a través de redes digitales que abarcan toda la organización o conectan varias organizaciones. Estas empresas tienen la capacidad de detectar y responder rápidamente a su entorno, ofreciendo flexibilidad en términos de tiempo y espacio, y aprovechan la tecnología para integrar de manera estrecha a proveedores, clientes y empleados (Laudon & Laudon, 2012).

El éxito de un sistema de información no se limita a la eficiencia en términos de costos y recursos, sino que también se mide por su capacidad para apoyar las estrategias de negocio, mejorar la estructura y cultura organizacional, y aumentar el valor para los clientes y el negocio de la empresa. Sin embargo, la mala administración y aplicación de la tecnología y los sistemas de información pueden generar problemas de desempeño que resultan en fallas tecnológicas y empresariales (O'Brien & Marakas, 2006).

#### 2.3.4.6. Base de datos

Una base de datos es una colección de datos organizada en un formato estructurado. El modelo relacional es el más utilizado, debido a que es independiente de la aplicación (móvil o de escritorio), es decir, que permite la modificación de datos sin afectar la estructura de la aplicación. Además, se basa en relaciones complejas entre tablas optimizando la velocidad de respuesta para la visualización, inserción o modificación de datos en una o varias tablas (Oppel & Sheldon, 2009). La Tabla 12 detalla la terminología relacionada con la administración de bases de datos.

**Tabla 12: Terminología sobre base de datos**

<b>Termino</b>	<b>Descripción</b>
Tabla	Implementación física de una relación en una base de datos relacional.
Relación	Representa la estructura que organiza los datos en una base de datos relacional.
Esquema	Descripción estructural y lógica de una base de datos, incluyendo tablas y relaciones.
Entidad	Representa un objeto o concepto del mundo real (persona, lugar, cosa, evento o actividad) sobre el cual los datos son recolectados.
Atributo	Hecho simple que describe o caracteriza una entidad de alguna manera.
Dominio	Define el tipo de datos que son almacenados en un atributo particular.

(continúa en la siguiente página)



(continuación)

Elemento	Descripción
Tupla	Conjunto de datos cuyos valores hacen una instancia de cada atributo definido por esa relación. Una tupla también es conocida como fila y representa un registro de datos relacionados.
Identificador	Atributo o conjunto de atributos que identifica de manera única a cada fila de datos en una relación.
Key (clave única)	Atributo que garantiza que existan registros únicos en cada tabla. Normalmente, el identificador es la clave única.
Normalización	Proceso de diseño de base de datos que reduce la redundancia y mejora la integridad.

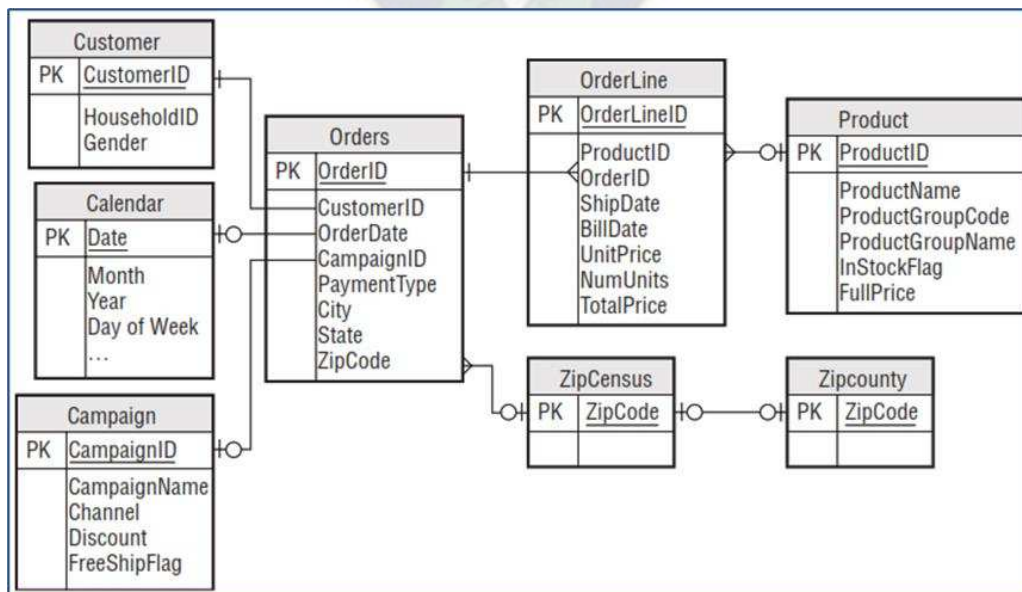
Fuente: Elaboración propia con base en Oppel & Sheldon (2009)

### 2.3.4.7. Modelo relacional

El modelo relacional de una base de datos consiste en un conjunto de columnas y filas reunidas en una estructura en forma de tabla que representa una entidad única formada por los datos relacionados.

Diferentes tablas se relacionan entre sí a través de claves únicas, lo que hace posible buscar información disponible en varias tablas de manera simultánea. Estas relaciones tienen una característica llamada cardinalidad, que es el número de elementos relacionados en cada lado de las tablas (Linoff, 2015). La Imagen 5 representa la relación de tablas donde los atributos que incluyen la terminación “ID” son las claves únicas de cada tabla.

Imagen 5: Representación del modelo relacional



Fuente: Linoff (Data Analysis Using SQL and Excel, 2015, pág. 7)

## 2.4. Herramientas específicas

### 2.4.1. Microsoft Excel

Microsoft Excel es un software informático de uso común para el análisis de datos. Utiliza un formato tabular y brinda a los usuarios la capacidad de modificar columnas y filas, aplicar funciones, crear gráficos, insertar tablas dinámicas y gran variedad de opciones para personalizar la interfaz (Linoff, 2015).

Este software es ampliamente utilizado por las empresas porque ofrece una amplia gama de características y funcionalidades que se detallan en la Tabla 13.

**Tabla 13: Características de Microsoft Excel**

<b>Característica</b>	<b>Descripción</b>
Hojas de cálculo	Conforman un libro de trabajo. Cada hoja está dispuesta como una tabla que contiene filas y columnas.
Celda	Intersección entre una fila y una columna de la hoja de cálculo. Las filas se denotan por números y las columnas por letras mayúsculas.
Rango	Conjunto de celdas contiguas (celda superior izquierda hasta la celda inferior derecha, por ejemplo, A1:A3 o A1:B4).
Funciones	Fórmulas predefinidas que permiten realizar cálculos (matemática, estadística, finanzas, lógica, etc.) con los datos de las hojas de cálculo.
Tablas dinámicas	Herramienta de análisis de datos que permite resumir y visualizar grandes cantidades de información de una manera fácil y rápida.
Macros	Programas automatizados que se crean en Excel para realizar tareas repetitivas de manera más eficiente.
Validación de datos	Permite controlar y restringir los valores que se pueden ingresar en una celda o rango de celdas. Esto permite mejorar la calidad de los datos y evitar errores en la información ingresada.
Gráficos	Forma visual de representar los datos.
Formas	Figura geométrica a la que se le puede asignar un procedimiento de automatización.
Protección de hojas y libros	Permite limitar el acceso a los datos y proteger la información de cambios no autorizados. Es posible establecer contraseñas y permisos de usuario para controlar el acceso a la información.

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de:

<https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

## 2.4.2. VBA

VBA (Visual Basic for Applications) es un lenguaje de programación que se basa en el paradigma de programación estructurada. Sin embargo, también admite algunos elementos del paradigma de programación orientada a objetos, como clases y objetos. Se utiliza para desarrollar aplicaciones o complementos para las aplicaciones de Microsoft Office. (Amelot, 2016)

Este lenguaje de programación tiene varias características favorables para su uso, como la facilidad de aprendizaje con una amplia comunidad de usuarios y numerosos recursos disponibles en línea. Permite desarrollar aplicaciones personalizadas en Office y automatizar tareas en estas aplicaciones. Además, ofrece herramientas integradas para encontrar y corregir errores en el código. La Tabla 14 detalla estas características.

**Tabla 14: Características del lenguaje VBA**

<b>Característica</b>	<b>Descripción</b>
Facilidad	Lenguaje de programación de fácil aprendizaje que cuenta con una amplia comunidad de usuarios y gran cantidad de recursos disponibles en la web.
Integración	VBA permite desarrollar aplicaciones personalizadas de acuerdo con las necesidades específicas de los usuarios.
Accesibilidad	VBA tiene acceso completo a los objetos de Office, lo que permite automatizar prácticamente cualquier tarea en las aplicaciones de Microsoft Office.
Depuración	VBA tiene herramientas para encontrar y corregir errores en el código.

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de:

<https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

El lenguaje VBA también tiene varias características técnicas notables. En primer lugar, ofrece una amplia variedad de procedimientos (Ver Tabla 15), tipos de datos (ver Tabla 17), instrucciones y estructuras de control (Ver Tabla 16). Además, el lenguaje VBA ofrece la capacidad de manipular y controlar objetos en las aplicaciones de Microsoft Office, lo que brinda la posibilidad de automatizar tareas y personalizar el comportamiento de dichas aplicaciones.

**Tabla 15: Procedimientos del lenguaje VBA**

<b>Procedimiento</b>	<b>Descripción</b>
Sub	Realiza una serie de operaciones en un bloque de código. Su ejecución se puede asignar a un botón o a un evento, también puede ser llamado por el nombre desde otra subrutina.
Function	Devuelve un valor y puede ser llamado por el nombre desde otra función o subrutina.
Property Get	Se utiliza para obtener el valor de una propiedad.
Property Let	Se utiliza para establecer el valor de una propiedad.

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de:  
<https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

**Tabla 16: Instrucciones del lenguaje VBA**

<b>Instrucción</b>	<b>Descripción</b>
Dim	Declara variables y les asigna espacio de almacenamiento.
If	Ejecuta condicionalmente un grupo de instrucciones, dependiendo del valor de una expresión.
Select Case	Ejecuta uno de varios grupos de instrucciones, dependiendo del valor de una expresión.
Do Loop	Repite un bloque de instrucciones cuando una condición es True o hasta que una condición se convierta en True.
For Next	Repite un grupo de instrucciones, un número especificado de veces.
For Each	Repite un grupo de instrucciones para cada elemento de un conjunto de objetos o una matriz.
Set	Asigna una referencia de objeto a una variable o propiedad.

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de:  
<https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

**Tabla 17: Tipos de dato**

<b>Tipo</b>	<b>Descripción</b>
Boolean	Representan valores lógicos verdadero o falso.
Integer	Números enteros sin decimales. Pueden ser positivos o negativos.
Date	Representan fechas y horas.
String	Conjuntos de caracteres que representan texto.
Variant	Puede contener cualquier tipo de dato

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de: <https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

### 2.4.3. SQL

SQL (Structured Query Language) es un lenguaje de consulta utilizado para administrar bases de datos relacionales. Permite a los usuarios crear, modificar y consultar bases de datos. También permite definir relaciones entre los datos y establecer restricciones para garantizar la integridad de los datos. La Tabla 18 detalla algunas características y sintaxis de SQL que usa muchos de los elementos que la sintaxis de VBA.

**Tabla 18: Características de SQL**

<b>Termino</b>	<b>Descripción</b>
Consulta	Lenguaje estructurado para la inserción, extracción y modificación de datos en una tabla.
Tabla	Objeto de la base de datos que contiene los datos organizados en filas (registros) y columnas (campos).
Registro	Representación de un conjunto de datos ubicados en una misma fila.
Campo	Cualquier columna dentro de una tabla, la cual tiene un tipo de dato y una longitud específica.
Dato	Representación simbólica de un atributo.
Select	Instrucción que especifica los datos que devolverá la consulta.
Insert	Instrucción que agrega uno o más registros a una sola tabla.

(Continúa en la siguiente página)



(continuación)

<b>Termino</b>	<b>Descripción</b>
Termino	Descripción
Update	Instrucción que modifica los valores de un conjunto de registros existentes en una tabla.
ODBC	(Open Database Connectivity) tecnología que permite a Microsoft Excel acceder a una base de datos.
GETDATE	Función que devuelve la fecha actual en la misma zona horaria que el sistema operativo.

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de:

<https://learn.microsoft.com/es-es/sql/sql-server/?view=sql-server-2016>

#### **2.4.4. SAP**

SAP es una empresa multinacional alemana fundada en 1972. El software desarrollado por SAP se convirtió en el estándar global de planificación de recursos empresariales (ERP<sup>1</sup>). Actualmente, SAP S/4HANA utiliza la tecnología de cómputo en memoria para procesar grandes volúmenes de datos y admitir tecnologías avanzadas como la inteligencia artificial (IA) y el aprendizaje automático. El amplio catálogo de soluciones de software de SAP se adapta a empresas de cualquier tamaño, permitiendo el mapeo y diseño de procesos empresariales, la recopilación y procesamiento de datos en una plataforma unificada, y la integración de datos operativos y de experiencia del cliente. Además, las soluciones de SAP ofrecen flexibilidad de implementación, ya sea en entornos locales o en la nube, para optimizar el análisis y el diseño eficiente de toda la cadena de valor. (SAP, 2023)

SAP HANA es una plataforma de datos que permite ejecutar análisis avanzados y transacciones de alta velocidad. Además de ser una base de datos, SAP HANA ofrece capacidades avanzadas de búsqueda, análisis e integración de datos estructurados y no estructurados. También funciona como un servidor de aplicaciones y permite la creación de aplicaciones inteligentes basadas en datos en tiempo real. (SAP, 2023)

En la Tabla 19 se detallan varios términos del sistema SAP que están relacionados con el caso de estudio de la memoria laboral. Aunque estos términos no están directamente vinculados al desarrollo del formulario automatizado, son relevantes para comprender el contexto del entorno laboral. Proporcionan una base de conocimiento adicional que puede ayudar a tener una visión más completa y contextualizada.

---

<sup>1</sup> Enterprise Resource Planning

**Tabla 19: Términos del sistema SAP**

<b>Término</b>	<b>Descripción</b>
Usuario	Objeto de base de datos que identifica a una persona para acceder o conectarse a una base de datos.
Función	Unidad lógica de trabajo o un conjunto de tareas específicas que se pueden ejecutar en el sistema. Las funciones están asociadas a los diferentes módulos funcionales de SAP.
Perfil	Conjunto de parámetros y configuraciones que definen cómo se comporta el sistema para un usuario o un conjunto de usuarios.
Rol	Conjunto predefinido de autorizaciones y permisos que se asignan a un usuario o a un grupo de usuarios. Un rol se crea de acuerdo con las necesidades y responsabilidades del usuario en la organización y determina qué acciones y funciones específicas puede realizar en el sistema.
Conflicto	Situación donde las autorizaciones y permisos asignados a los usuarios generan riesgos para la seguridad, integridad o eficiencia de los procesos empresariales (autorización, segregación de funciones, conflictos de datos y conflictos de configuración).
Gestión de roles	Proceso de monitoreo y prevención de riesgos durante la creación y actualización de roles. Garantiza que los usuarios tengan las autorizaciones adecuadas para realizar sus tareas de manera segura y eficiente, al tiempo que se mantienen los controles y la segregación de funciones necesarios.
Segregación de funciones	Asignación de roles y responsabilidades dentro del sistema SAP de manera que haya una separación adecuada de funciones y actividades que minimice los riesgos y prevenga posibles conflictos de interés o fraudes.
ABAP	(Advanced Business Application Programming) Lenguaje de programación utilizado en el entorno SAP para desarrollar programas que manejan datos y ejecutan diversas operaciones, como el acceso a bases de datos, el control del flujo del programa y la modularización del código.

Fuente: Elaboración propia con base en la documentación de SAP (2023)

## **2.5. Personalización**

Crear software personalizado es una opción viable cuando no se encuentra disponible un software comercial adecuado para una tarea deseada o cuando el costo de adquirirlo es elevado. Además, las organizaciones que buscan obtener una ventaja competitiva a través de sistemas de información innovadores o que tienen requerimientos especializados pueden beneficiarse de desarrollar su propio software. Esto les permite responder a necesidades específicas de usuarios y empresas, además de tener control sobre el mantenimiento del software. Sin

embargo, es importante considerar que la creación de software personalizado puede implicar un costo inicial considerablemente alto y la necesidad de contar con un equipo de desarrollo competente. También se asume la responsabilidad del mantenimiento continuo como creador del software (Kendall & Kendall, 2011).

El desarrollo personalizado de add-ins permite a los usuarios automatizar tareas rutinarias y agregar nuevas funciones adaptadas a sus necesidades específicas. De esta manera, se pueden aprovechar las capacidades de las herramientas de Microsoft Office para optimizar los procesos empresariales y mejorar la productividad. (Microsoft, 2023)

La personalización de las aplicaciones de Microsoft Office puede realizarse mediante complementos (add-ins) desarrollados con Visual Studio, que agregan funcionalidades adicionales a las aplicaciones de Office, como botones personalizados, paneles de tareas o integraciones con servicios externos. Otro tipo de proyecto es la automatización de Office mediante macros, utilizando el lenguaje VBA. Estas macros permiten crear scripts personalizados para realizar tareas repetitivas de manera automatizada. (Microsoft, 2023)

Existen dos tipos de proyecto para personalizar las aplicaciones de Microsoft Office:

- 1) Personalización a nivel de documento donde el código se ejecuta únicamente cuando se abre un documento específico.
- 2) Personalización a nivel de aplicación donde el código está disponible cada vez que se ejecuta la aplicación.

La Tabla 20 proporciona una visión general y simplificada de las características, diferencias y similitudes entre la personalización con VBA y Visual Studio.

**Tabla 20: Comparación de la personalización con VBA y Visual Studio**

<b>Criterio</b>	<b>VBA</b>	<b>Visual Studio</b>
Lenguaje	VBA	C#, VB.NET, C++
Código	Está conectado a un documento concreto y se conserva en él.	Se almacena por separado del documento
Tipo de aplicaciones	Aplicaciones de Office	Amplio rango de aplicaciones, desde escritorio hasta web
IDE	Integrado en Office	Independiente de Office
Acceso	Objetos de Office y API de VBA	Objetos de Office, API y servicios de .NET Framework
Integración	Aplicaciones de Office	Aplicaciones Office, diversas tecnologías y plataformas

Fuente: Elaboración propia con base en la documentación de Microsoft Office.



## 2.6. Ética en el desarrollo de software

Según Pressman (2010) los profesionales en el desarrollo de software deben acatar un código de ética y deben conducirse de acuerdo con las siguientes reglas:

- Nunca robar datos.
- Nunca distribuir o vender información con derechos de autor.
- Nunca destruir o modificar maliciosamente programas, archivos o datos.
- Nunca violar la privacidad de un individuo, grupo u organización.
- Nunca hackear un sistema por beneficio.
- Nunca crear o propagar un programa maligno.
- Nunca usar tecnología para facilitar la discriminación o el hostigamiento.



## Capítulo 3: Diagnóstico y Propuesta

El enfoque de procesos resulta fundamental en el diagnóstico de la situación actual de la empresa en relación con la ineficiencia en el llenado del formulario para el procedimiento de “Gestión de riesgos para el alta o modificación de roles de usuarios” y la propuesta de automatización de este. Este enfoque se basa en la identificación, comprensión y gestión de los procesos interrelacionados de la organización, permitiendo controlar la interacción entre ellos y mejorar el desempeño global. Implica definir y gestionar de manera sistemática los procesos y sus interacciones para alcanzar los resultados previstos (International Organization for Standardization [ISO], 2015).

La identificación significativa de los procesos puede ser un desafío debido a la naturaleza jerárquica de los procesos empresariales. En el contexto del diagnóstico, es necesario realizar dos fases clave: la designación y la evaluación de los procesos. La designación busca comprender los procesos y sus interrelaciones, mientras que la evaluación prioriza aquellos procesos que requerirán acciones de gestión (Dumas, La Rosa, Mendling, & Reijers, 2013).

En el caso de SOBOCE S.A., la fase equivalente a la designación de procesos se denomina "identificación de la información del proceso". Esta etapa se aplica específicamente cuando se producen cambios o se crea un nuevo proceso. Para llevar a cabo esta identificación, se realiza un relevamiento exhaustivo de información, validación, análisis y revisión de las actividades correspondientes.

### 3.1. Diagnóstico

#### 3.1.1. Situación de la empresa

La adquisición mayoritaria de las acciones de SOBOCE S.A. por parte de YURA Inversiones Bolivia S.A., perteneciente al grupo GLORIA, implicó una migración del sistema ERP<sup>2</sup> de JD Edwards a SAP HANA. Este cambio permitió la integración y optimización de los procesos de negocio mediante el análisis en tiempo real de grandes volúmenes de datos. Sin embargo, la implementación del sistema SAP enfrentó diversos desafíos, entre los cuales se destacan la definición de roles de usuario y la correcta segregación de funciones para mitigar los riesgos asociados a la seguridad y la integridad de los datos.

En este sentido, se identificaron riesgos potenciales, como el acceso no autorizado a información sensible, la manipulación indebida de datos y las posibles violaciones de la

---

<sup>2</sup> Enterprise Resource Planning

privacidad. Ante esta situación, la dirección de SOBOCE S.A. estableció un procedimiento de gestión de riesgos que involucró el análisis detallado de datos, la adecuada definición y asignación de roles de usuario en SAP, la implementación de controles y la generación de registros mediante formularios.

Para abordar la necesidad de generar los formularios, la opción más adecuada es utilizar el lenguaje ABAP y desarrollar reportes personalizados. Sin embargo, esta alternativa no era viable para la empresa en ese momento debido a la falta de personal con las habilidades y conocimientos requeridos en el lenguaje ABAP, además del tiempo limitado disponible para capacitar al equipo de “Soluciones y Proyectos”. La contratación de consultores en desarrollo de soluciones SAP representaba costos significativos, como se evidencia en la Tabla 21.

**Tabla 21: Costo de desarrolladores de soluciones SAP**

<b>Página Web</b>	<b>USD/h</b>
<a href="http://www.workana.com">www.workana.com</a>	18,00
<a href="http://www.workana.com">www.workana.com</a>	20,00
<a href="http://mx.talent.com">mx.talent.com</a>	73,85
<a href="http://www.freelancermap.com">www.freelancermap.com</a>	81,00

Fuente: Elaboración propia con base a la información de páginas web

Ante esta situación, se buscó una alternativa viable que permitiera cumplir de manera eficiente los requerimientos del procedimiento. La alternativa propuesta por el autor consistía en utilizar el lenguaje VBA para automatizar las tareas repetitivas del registro de datos en el formulario.

### **3.1.1.1. Lineamiento para el desarrollo de software**

La automatización mediante Software es bastante útil para optimizar los procesos de una empresa. Sin embargo, es importante que todo desarrollo de software esté alineado a los procesos estratégicos de la empresa.

La política de SOBOCE sobre el desarrollo de software hace referencia a lo siguiente:

1. Todo desarrollo de aplicaciones de software debe estar alineado a los procesos estratégicos de la empresa.
2. Todo desarrollo de aplicaciones de software en la empresa puede ser adquirido de terceras partes o desarrolladas por personal propio, en función a la evaluación de la necesidad.
3. El área de Tecnología y Procesos debe elaborar, mantener y actualizar procedimientos documentados, en los cuales se determinen los lineamientos, procesos, buenas prácticas, plantillas y demás medios que sirvan para regular el desarrollo de software interno y externo en un ambiente de mitigación del riesgo y aseguramiento de la calidad.

### **3.1.2. Identificación de la información del proceso**

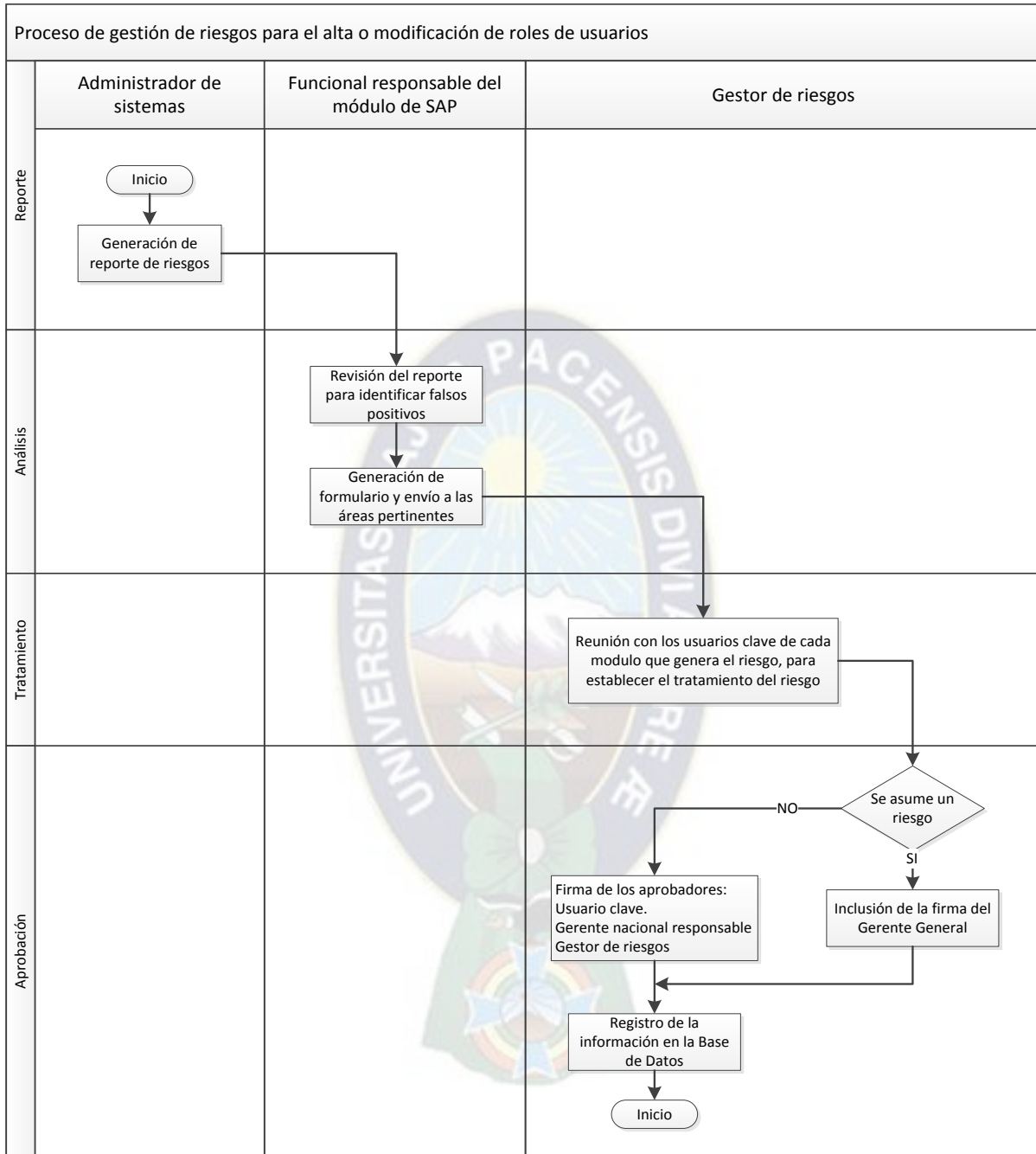
En el contexto de la propuesta de automatización del formulario para el procedimiento de "Gestión de riesgos para el alta o modificación de roles de usuarios", y como parte del diagnóstico, se realizó un análisis exhaustivo de la información del proceso resultante de la revisión de documentación y reuniones con los responsables. Como resultado, se creó un diagrama de flujo de funciones cruzadas que detalla las actividades y los responsables, así como un diagrama de flujo del proceso para identificar las actividades a automatizar.

#### **3.1.2.1. Diagrama de flujo de funciones cruzadas**

El diagrama de flujo de funciones cruzadas muestra la relación entre un proceso empresarial y las unidades funcionales responsables de dicho proceso (Microsoft, 2021). Este diagrama proporciona una visión integral de cómo se interconectan las actividades y los responsables, lo cual facilita la identificación de los requerimientos generales para automatizar el formulario del procedimiento. La Imagen 6 detalla el diagrama de flujo de funciones cruzadas del proceso.



**Imagen 6: Diagrama de flujo de funciones cruzadas**









Fuente: Elaboración propia con base en la documentación de SOBOCE S.A.

### 3.1.2.2. Diagrama de flujo del proceso

El diagrama de flujo del proceso proporciona un nivel de detalle más exhaustivo y resulta especialmente útil para identificar periodos no productivos, como desplazamientos, demoras y almacenamiento temporal. Al identificar estos períodos improductivos, los analistas pueden tomar medidas para minimizarlos y reducir los costos asociados (Niebel & Freivalds, 2009).

Se adaptó el diagrama de flujo del proceso para identificar tiempos prolongados en las actividades y visualizar cuellos de botella y tareas susceptibles de ser automatizadas. La Tabla 22 describe los símbolos utilizados para representar un diagrama de flujo de procesos donde se procesan datos y se obtiene un formulario aprobado.

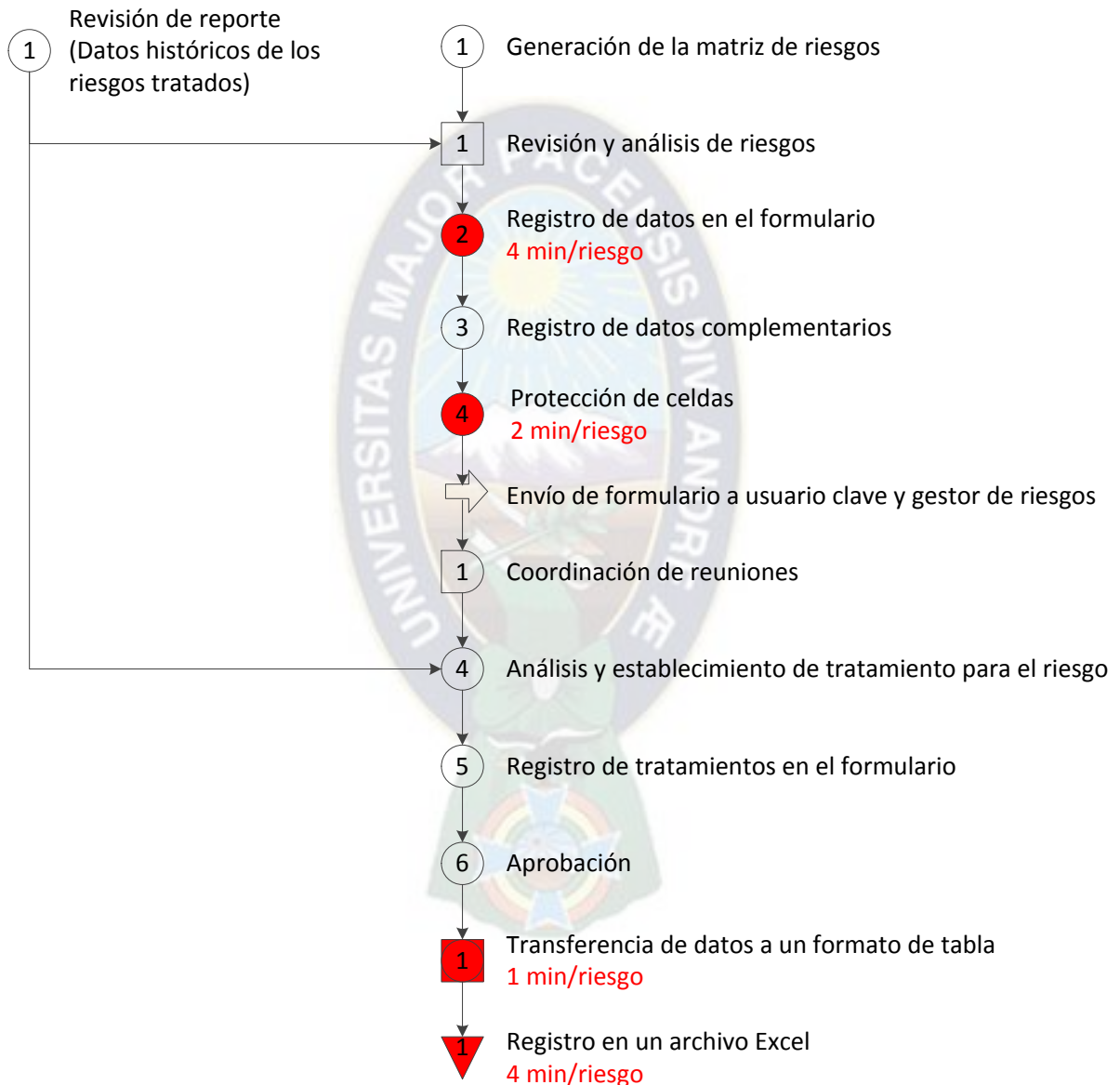
**Tabla 22: Símbolos del diagrama de flujo del proceso**

Símbolo	Actividad	Descripción
	Operación	Acción realizada para transformar un conjunto de datos.
	Inspección	Revisión de la calidad/cantidad del conjunto de datos por parte de uno o más responsables.
	Actividad combinada	Acciones de operación e inspección por un mismo responsable.
	Demora	Tiempo de espera o demora en el flujo del conjunto de datos.
	Almacenamiento	Registro en una base de datos.
	Envío	El conjunto de datos se transfiere de un responsable a otro.

Fuente: Elaboración propia con base en Niebel & Freivalds (2009)

La Imagen 7 detalla el diagrama de flujo del proceso antes de la implementación del formulario automatizado. En este diagrama se identifican de color rojo las operaciones repetitivas y el tiempo aproximado de su realización. Estas operaciones, al ser repetitivas y no requerir análisis, son candidatas ideales para la automatización.

**Imagen 7: Diagrama de flujo del proceso sin la automatización**



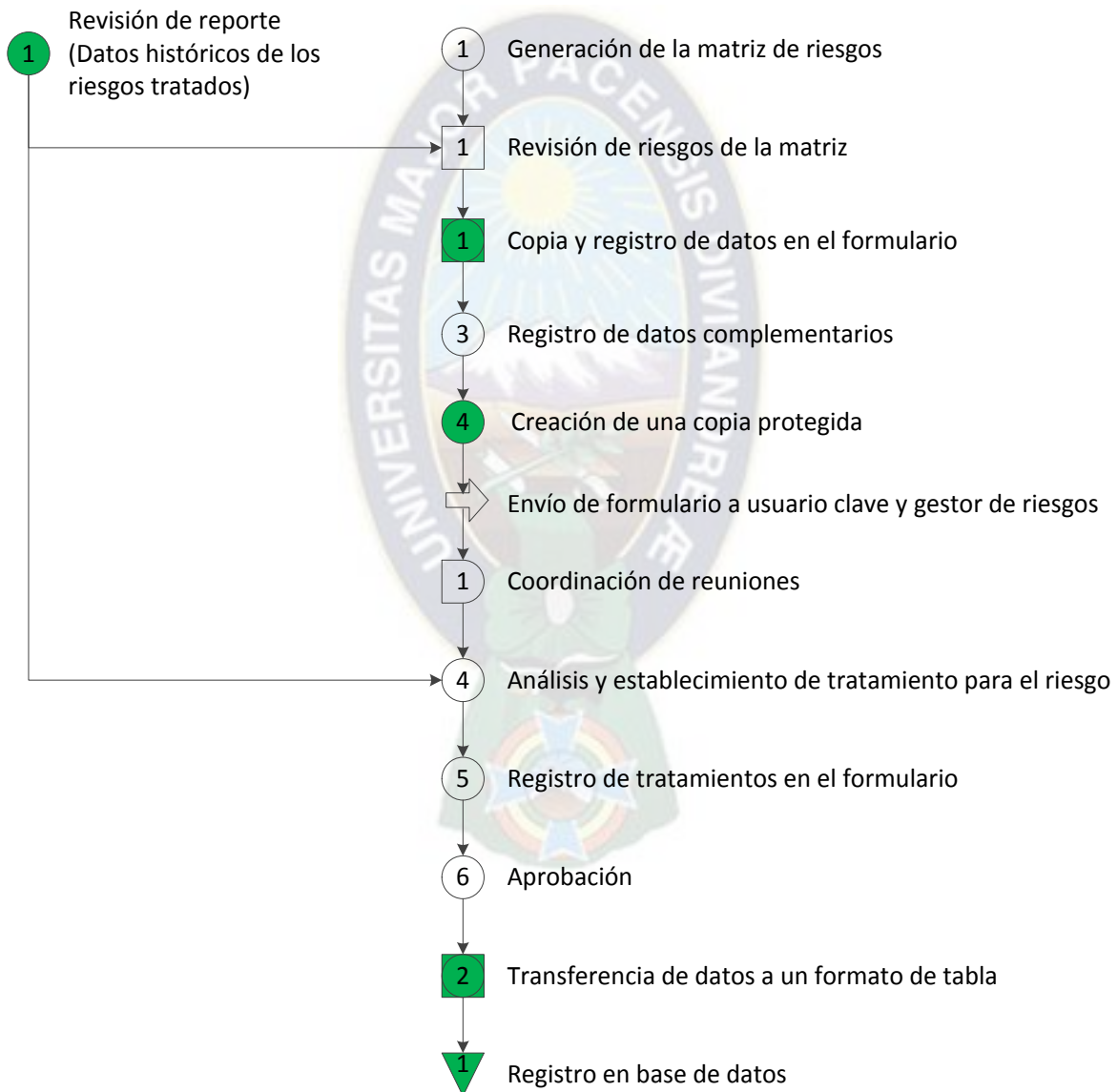
Fuente: Elaboración propia con base en la documentación de SOBOCE S.A.



### 3.2. Propuesta

La Imagen 8 detalla el diagrama de flujo del proceso propuesto. Es importante destacar que si bien el diagrama no presenta cambios significativos con respecto al diagrama de flujo inicial, esto se debe a que las tareas del proceso siguen un orden secuencial. Lo importante de la implementación de la automatización no se centra en reducir tareas si no eliminar la necesidad de realizar manualmente estas tareas y minimizan los errores humanos.

**Imagen 8: Diagrama de flujo del proceso propuesto**



Fuente: Elaboración propia con base en la documentación de SOBOCE S.A.

### 3.2.1. Requerimientos

La Tabla 23 describe los requerimientos funcionales para el desarrollo del formulario automatizado.

**Tabla 23: Requerimientos de desarrollo**

<b>Requerimiento</b>	<b>Descripción</b>
Copia de datos.	Funcionalidad para la copia de datos desde un reporte al formulario.
Aplicación de Formato	Funcionalidad para aplicar el formato requerido al conjunto de datos copiado desde el reporte.
Creación de una copia protegida	Funcionalidad para la creación de una copia protegida del formulario para evitar la modificación de datos. La extensión de la copia debe ser xlsx.
Transferencia de datos	Funcionalidad para transferir los datos desde la copia creada al formulario (después del registro de los tratamientos y aprobaciones correspondientes).
Registro en base de datos	Funcionalidad de conexión con el servidor de SOBOCE S.A. para registrar los datos en una base de datos SQL.
Modificación de la base de datos	Funcionalidad para la modificación en la base de datos para finalizar los riesgos que se hayan eliminado o finalizar el riesgo debido a un cambio o baja del usuario que genera el riesgo.
Generación de reporte	Funcionalidad para generar un reporte desde la base de datos.

Fuente: Elaboración Propia.

Implícitamente, todo desarrollo de Software debe contar con requerimientos técnicos adecuados para garantizar la calidad en el desarrollo. La Tabla 24 detalla los requerimientos técnicos necesarios para la construcción de un software eficiente y efectivo.

**Tabla 24: Requerimientos técnicos**

<b>Requerimiento</b>	<b>Descripción</b>
Diseño de la arquitectura	Detalle de los diferentes niveles del software
Rendimiento	<ul style="list-style-type: none"><li>• Velocidad de respuesta</li><li>• Escalabilidad</li><li>• Uso eficiente de los recursos del sistema.</li></ul>
Interfaz de usuario	Interfaz intuitiva para la interacción eficiente y efectiva de los usuarios.
Integración	<ul style="list-style-type: none"><li>• IDE de desarrollo de Microsoft.</li><li>• Compatibilidad con diferentes versiones de Excel.</li><li>• Pruebas exhaustivas con diferentes versiones de Excel</li></ul>
Mantenimiento	<ul style="list-style-type: none"><li>• Patrones de diseño comunes y estructura en módulos.</li><li>• Documentación detallada del código.</li></ul>

Fuente: Elaboración Propia con base en Pressman (2010).



## Capítulo 4: Marco Aplicativo

El modelado es una técnica ampliamente utilizada en la ciencia y la ingeniería para proporcionar abstracciones de sistemas con un cierto nivel de precisión y detalle. Una vez creado el modelo, este se analiza para obtener una comprensión más profunda del sistema en cuestión. En el diseño y desarrollo de software basado en modelos, el modelado de software se convierte en una parte esencial del proceso de desarrollo. Los modelos se construyen y analizan antes de la implementación del sistema y se utilizan para guiar su posterior implementación. (Cervantes & Kazman, 2016)

### 4.1. Diseño

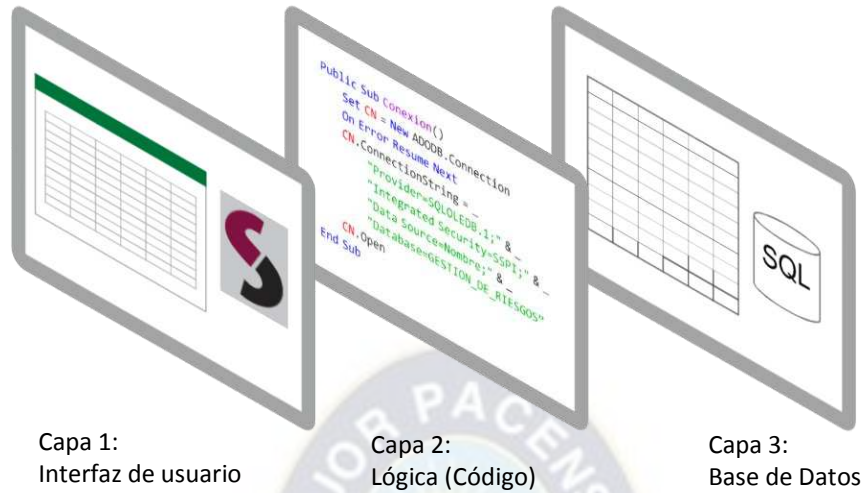
La arquitectura de software se puede describir en diferentes niveles de detalle, desde una descripción más general hasta una descripción más específica. En un nivel alto, puede describir la descomposición del sistema de software en subsistemas. En un nivel inferior, puede describir la descomposición de subsistemas en módulos o componentes. En ambos niveles de descripción, se hace énfasis en la vista externa del subsistema o componente. De esta manera, la arquitectura de software puede ser vista como un conjunto de subsistemas y componentes que trabajan juntos de manera interconectada y coordinada para lograr la funcionalidad general del sistema. (Cervantes & Kazman, 2016)

#### 4.1.1. Arquitectura

La arquitectura basada en capas es un patrón de diseño con la idea de separar la lógica del negocio de la lógica de presentación y acceso a datos, para permitir una mayor flexibilidad y escalabilidad del sistema. El sistema se divide en capas, donde cada capa es independiente con una funcionalidad específica y se comunica con las capas adyacentes (Cervantes & Kazman, 2016).

En el contexto del formulario automatizado, se ha empleado la arquitectura de capas como base en el desarrollo para garantizar un diseño eficiente y modular (ver Imagen 9). Aunque no el formulario no ha sido desarrollado en su totalidad, porque se aprovechó la interfaz de Microsoft Excel. La Tabla 25 detalla las características de cada una de las capas del formulario.

**Imagen 9: Capas del formulario automatizado**



Fuente: Elaboración propia.

**Tabla 25: Arquitectura de capas del formulario automatizado**

Capa	Descripción	Características del formulario automatizado
Interfaz	Se encarga de la presentación de la información al usuario.	<ul style="list-style-type: none"> <li>• Interfaz intuitiva.</li> <li>• Campos de entrada de datos</li> <li>• Botones</li> <li>• Tabla de visualización de datos</li> </ul>
Lógica	Contiene la lógica donde se procesan los datos.	<ul style="list-style-type: none"> <li>• Validación de datos</li> <li>• Actualización de datos</li> <li>• Procesamiento de datos (copia, concatenación, transferencia)</li> <li>• Generación de consultas SQL</li> </ul>
Base de datos	Realiza operaciones de lectura y escritura de datos	<ul style="list-style-type: none"> <li>• Registro</li> <li>• Modificación</li> <li>• Recuperación para reportes</li> </ul>

Fuente: Elaboración propia con base en Pressman (2010)

#### 4.1.2. Interfaz

En el mundo actual la usabilidad de los productos de alta tecnología es crucial para su éxito. La usabilidad se refiere a la facilidad y eficiencia con la que los usuarios interactúan con las funciones y características de un producto. Una interfaz bien diseñada permite a los usuarios utilizar el software sin esfuerzo, mientras que una interfaz mal concebida puede generar frustración y baja eficiencia en el trabajo. (Pressman, 2010).

En este sentido, El diseño de la interfaz del formulario automatizado se centra en mejorar la usabilidad y la eficiencia del registro. Para esto se han implementado características como listas desplegables para facilitar la selección de datos y funciones de formato condicional que guían al usuario durante el ingreso de información. Además, la interfaz se ha desarrollado siguiendo los lineamientos del sistema integrado de SOBOCE y se basa en una plantilla estándar para documentos de registro, lo que garantiza la coherencia y consistencia en la captura de datos. El formulario se divide en cuatro secciones claramente identificadas: Datos generales, Usuarios, Riesgos y Aprobadores.

La sección de datos generales (ver Imagen 10) está diseñada para los datos del solicitante y los datos de referencia que permiten identificar al formulario.

**Imagen 10: Sección de datos generales**

FECHA DE PUBLICACIÓN DEL FORMATO:		Nº DE REVISIÓN DEL FORMATO:	
Número de Ticket: <small>(Llenado solo por Service Desk)</small>	<b>REGISTRAR NÚMERO DE TICKET</b>	Referencia 1	
		Referencia 2	
EMPRESA:		Fecha:	
		14/04/2023	
Nombre Solicitante (Líder de Módulo)		Cargo	Módulo
			Área

Fuente: Manual de uso elaborado para SOBOCE S.A.

La sección de usuarios (ver Imagen 11) está diseñada para los códigos de usuario SAP para los cuales los roles asignados generan los riesgos. Además incluye una sección para las observaciones generales relacionadas con los usuarios.

**Imagen 11: Sección de usuarios**

<b>1. USUARIO(S)</b>								
1)		6)		11)		16)		Observaciones:  
2)		7)		12)		17)		
3)		8)		13)		18)		
4)		9)		14)		19)		
5)		10)		15)		20)		

Fuente: Manual de uso elaborado para SOBOCE S.A.



La sección de riesgos (ver Imagen 12) está diseñada para los datos relacionados con los riesgos generados por los roles de usuario y las acciones para tratar los mismos.

**Imagen 12: Sección de riesgos**

2. MITIGACIÓN DE RIESGOS					
1) ID-Riesgo:	Valoración:	Tratamiento:	Tipo de evidencia:	Periodicidad:	Vigencia hasta:
Descripción Riesgo:		Control de Mitigación:		Cargo Responsable:	
Función 1:		Función 2:		Función 3:	
Descripción función 1:		Descripción función 2:		Descripción función 3:	
Transacciones función 1:		Transacciones función 2:		Transacciones función 3:	

Fuente: Manual de uso elaborado para SOBOCE S.A.

La sección de aprobadores (ver Imagen 13) está diseñada para los datos de los aprobadores que correspondan al área donde se generan los riesgos. La aprobación también depende del tipo de tratamiento es decir el asumir un riesgo requiere un nivel más alto de aprobación que una medida de control.

**Imagen 13: Sección de aprobadores**

3. APROBACIONES		
USUARIO CLAVE	GERENCIA NACIONAL RESPONSABLE	GESTOR DE RIESGOS SOBOCE
Fecha:	Fecha:	Fecha:
	GERENTE GENERAL SOBOCE	
	Fecha:	

Fuente: Manual de uso elaborado para SOBOCE S.A.

Una descripción detallada de los datos y el formato del formulario, se proporciona información adicional en el Anexo 2: Formulario (pág. 76)

#### 4.1.3. Algoritmos






Para desarrollar un complemento en Excel adecuado para el caso de estudio, es necesario utilizar algoritmos que sigan las características clave establecidas por Knuth (1997) y Cormen *et*



al. (2022). Es decir que los algoritmos deben ser finitos y deben estar definidos de manera precisa. La especificación de los algoritmos puede realizarse mediante diferentes medios, como un lenguaje de programación o diagramas de flujo, siempre y cuando proporcionen una descripción precisa del procedimiento a seguir.

Los algoritmos pueden ser representados visualmente a través de un diagrama de flujo. Este diagrama proporciona una representación gráfica de los pasos a seguir en un proceso (Pantigoso, 2006). Para representarlos gráficamente, se utiliza el estándar establecido por la norma ISO 5807, la Tabla 26 define los símbolos y convenciones utilizados en los diagramas de flujo. En el caso del formulario automatizado, los algoritmos diseñados para su funcionamiento se describen en detalle en el Anexo 3 (pág. 78).

**Tabla 26: Símbolos de un diagrama de flujo**

Símbolo	Nombre	Descripción
	Terminal	Se utiliza para indicar el inicio o fin de un diagrama de flujo.
	Entrada/Salida	Representa la entrada o salida de datos en el proceso.
	Proceso	Representa una operación que origina cambios en los datos.
	Decisión	Representa operaciones lógicas para la elección de una alternativa.
	Salida	Se utiliza para mostrar un resultado al usuario.

Fuente: Elaboración propia con base en la norma ISO 5807 (1985)

## 4.2. Construcción

### 4.2.1. Código

El código limpio es concreto, donde cada función, cada clase y cada módulo muestran una única actitud que se mantiene invariable y no se contamina por los detalles circundantes. Además, un código limpio implica que el programador ha dedicado tiempo y le ha dado importancia a los detalles para que sea sencillo y fácil de entender (Martin, 2009).

El uso disciplinado de varias técnicas probadas y ampliamente aplicadas es fundamental para generar código limpio. Estas técnicas se detallan en la Tabla 27 y fueron empleadas en el desarrollo del formulario automatizado.

**Tabla 27: Técnicas de código limpio**

<b>Técnica</b>	<b>Descripción</b>
Uso de Nombres descriptivos	Los nombres de una variable, función o clase deben ser descriptivos (por qué, qué hace y cómo se usa). Además, deben facilitar la búsqueda dentro del código o la documentación.
Modularidad y Abstracción	El código puede ser dividido en módulos separados y bien definidos, para abstraer las líneas de código que se repiten en varias partes en forma de una función o clase y luego llamar a esa función o clase desde cualquier módulo.
Indentación	Facilita la legibilidad y comprensión del código.
Uso de Comentarios	Permite explicar lo que no se puede expresar con en el código.
Control de errores	Es recomendable utilizar el manejo de excepciones (propio de cada lenguaje) al detectar un error, para especificar el comportamiento del programa en caso error.
Pruebas unitarias	Se enfocan en probar una función o un método para verificar que la unidad de código funciona correctamente.
Refactorización	Después de cada prueba se pueden hacer pequeñas modificaciones al código con el objetivo de eliminar la complejidad innecesaria y mejorar la legibilidad.

Fuente: Elaboración propia con base en Martin (2009)

#### **4.2.2. Entorno de desarrollo integrado**

El formulario automatizado se desarrolló en el entorno de desarrollo integrado (IDE) de Microsoft Office, esta herramienta proporciona un entorno de desarrollo eficiente y efectivo para la creación de complementos y aplicaciones utilizando el lenguaje VBA. Sus características de gestión de proyectos, sugerencias automáticas y capacidad de depuración mejoran la productividad del programador y contribuyen al desarrollo de soluciones de calidad. Estas características se detallan en la Tabla 28.

**Tabla 28: Características del editor de código de Microsoft Office**

<b>Característica</b>	<b>Descripción</b>
Gestión de Proyectos	El editor permite la creación y facilita la organización del código y los recursos asociados. Cuenta con un explorador de proyectos, vistas de código, ventana de propiedades, ventanas que muestran el valor de las variables y un examinador de objetos.
Módulo	Un módulo es un contenedor de procedimientos. Cada libro tiene asociado un proyecto que cuenta con módulos que se agrupan en cuatro categorías: <ol style="list-style-type: none"><li>1) Módulos asociados a objetos.</li><li>2) Módulos asociados a formularios (UserForm).</li><li>3) Módulos estándar (creación de procedimientos).</li><li>4) Módulos de clase. (creación de nuevos objetos)</li></ol>
Autocompletado	El editor sugiere nombres de objetos, métodos, propiedades y tipos de variables a medida que se escribe el código.
Depuración	El editor permite establecer puntos de interrupción, ver los valores de las variables en tiempo de ejecución, ejecutar el código paso a paso. También permite establecer si la ejecución del código se interrumpirá al ocurrir un error de ejecución.
Personalización	El editor permite modificar el estilo (color, tipo de letra, tamaño) de palabras clave, funciones, instrucciones, objetos, métodos, propiedades, comentarios y la identificación de errores.

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de: <https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

### **4.2.3. Codificación**

El código del formulario automatizado fue escrito en el IDE de Microsoft Excel, utilizando el lenguaje VBA y aplicando técnicas de código limpio. Este código fue escrito por el autor en su totalidad tomando como referencia información pública como la documentación de Microsoft, ejemplos de código en foros de internet y tutoriales. Por lo que la publicación del código no representa una filtración de la propiedad intelectual de SOBOCE S.A.

La presentación del código solo tiene fines académicos y no muestra información o datos que puedan afectar de alguna forma a la empresa. El código del formulario automatizado se describe y explica a detalle en el Anexo 4 (pág. 86).

### **4.3. Pruebas**

Los escenarios de prueba con datos se utilizan para simular diferentes situaciones y condiciones que el sistema podría enfrentar en el mundo real. Estos escenarios permiten mostrar y evaluar la

capacidad del sistema para manejar los datos de entradas y procesarlas de manera efectiva. La Imagen 14 es una representación de los datos que se obtienen del reporte del sistema SAP. Estos datos solo son de prueba y no reflejan la realidad de la empresa, su uso es solo con fines académicos.

**Imagen 14: Conjunto de datos de prueba**

USUARIO ID	RIESGO ID	NIVEL RIESGO	RIESGO DESCRIPCIÓN	FUNCION ID	FUNCION DESCRIPCIÓN	TRANSACCIÓN
UPRUEBA	ZSAR0022	Alto	Un usuario podría crear/modificar la factura de un cliente y luego ingresar/modificar los pagos hacia esa factura sin autorización.	ZFIFI004	Aplicacion de Caja	F-30
				ZFIFI005	Procesar Facturas de Clientes	F-22
	ZSAR0033	Alto	Un usuario podría ingresar/modificar las facturas de ventas y aprobar los limites de credito del cliente sin autorización.	ZFIAR003	Administracion de Creditos	FD32
				ZFIFI005	Procesar Facturas de Clientes	F-22
	ZSFI0049	Alto	El usuario podría registrar un pago no autorizado y luego realizar la Conciliacion bancaria sin autorización.	ZFIFI001	Pago a proveedores	F-48
				ZFIFI002	Procesar facturas de proveedores	F-43
						FB60
						MIRO
	ZSFI0067	Alto	Existe la posibilidad de crear un documento IP (forma interna de pago) y cancelar algun documento de cobro generando un saldo a favo	ZFIFI005	Procesar Facturas de Clientes	F-22
				ZFIFI006	Compensar saldo del Cliente	F-32
					FB1D	

Fuente: Manual de uso elaborado para SOBOCE S.A.

#### 4.3.1. Copia de datos

La prueba implica realizar la copia de datos desde el reporte al formulario, para esto son necesarios los datos de configuración inicial como la referencia de filas y columnas y los nombres del archivo desde donde se realizará la copia. La Tabla 29 detalla los parámetros de la prueba.

**Tabla 29: Parámetros de la prueba de copia de datos.**

Referencia	Valor
Responsable	Usuario (Analista de procesos)
Acción	Registro de datos
Datos de entrada	Número de fila, Nombre de columnas, Datos de riesgos.
Procedimiento	A_COPIAR_DATOS
Datos de salida	Datos de riesgos

Fuente: Elaboración propia.

El procedimiento de la prueba es el siguiente:

- 1) En la hoja “Configuración”, registro del número de fila del encabezado y el nombre de las columnas de la tabla que contiene los datos (ver Imagen 15).

**Imagen 15: Configuración**

USUARIO ID	RIESGO ID	NIVEL RIESGO	RIESGO DESCRIPCIÓN	FUNCION ID	FUNCION DESCRIPCIÓN	TRANSACCIÓN	REFERENCIA 1
UPRUEBA	ZSAR0022	Alto	Un usuario podr	ZFIF1004	Aplicacion de Caja	F-30	20220630
	ZSAR0033	Alto	Un usuario podr	ZFIF1005	Procesar Facturas	F-22	20220630
	ZSFI0049	Alto	El usuario podri	ZFIAR003	Administracion de	FD32	20220630
				ZFIF1005	Procesar Facturas	F-22	20220630
				ZFIF1001	Pago a proveedor	F-48	20220630
				ZFIF1002	Procesar facturas	F-43	20220630
						FB60	20220630
						MIRO	20220630
				ZFIF1005	Procesar Facturas	F-22	20220630
				ZFIF1006	Compensar saldo	F-32	20220630
				ZCOOC001	Administrar ciclo	FB1D	20220630
						KSU1	20220630
						KSU2	20220630
						KSU7	20220630
						KSU8	20220630
				ZCOOM002	Ejecutar distribuci	KSU5	20220630
						KSUB	20220630
				ZCOOC001	Administrar ciclo	KSU1	20220630
						KSU2	20220630
						KSU7	20220630
						KSU8	20220630
				ZCOOC003	Administrar Centr	KS01	20220630
						KS02	20220630

DATA	COLUMNA
USUARIO	B
ID RIESGO	C
VALORACIÓN	D
DESCRIPCIÓN RIESGO	E
FUNCIÓN	F
DESCRIPCIÓN FUNCIÓN	G
TRANSACCIÓN	H
REFERENCIA 1	I

Fuente: Manual de uso elaborado para SOBOCE S.A.

- 2) En la hoja “COPIA”, selección del nombre de archivo desde donde se copiarán los datos (ver Imagen 16).

**Imagen 16: Selección del nombre de archivo.**

Referencias al archivo del que se copiaran los da

Nombre de archivo

Nombre de hoja

PERSONAL.XLSB

XRT-BCLRP-001 FORMULARIO DE GESTIÓN DE RIESGOS PARA EL ALTA O MG

Matriz conflictos usuarios Soboce.xlsx

COPIAR DATOS

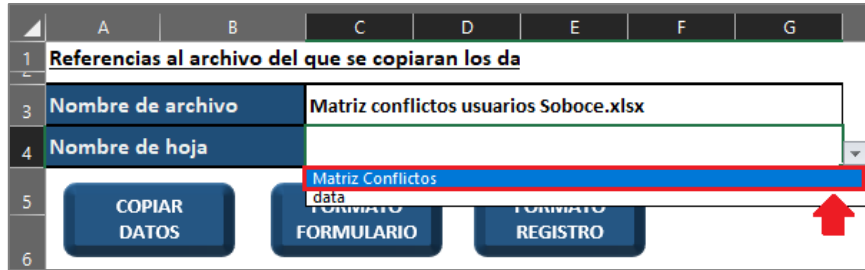
FORMATO FORMULARIO

FORMATO REGISTRO

Fuente: Manual de uso elaborado para SOBOCE S.A.

- 3) Seleccionar el nombre de la hoja que contiene los datos (ver Imagen 17).

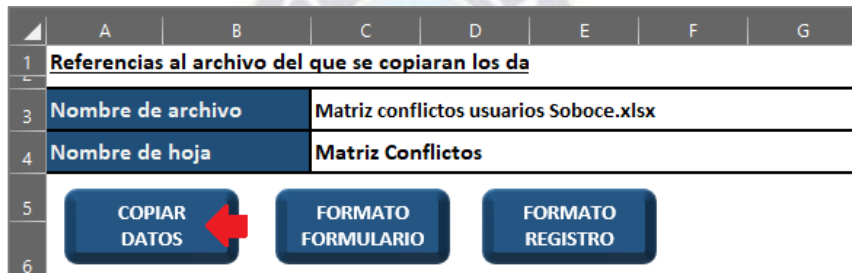
**Imagen 17: Selección del nombre de hoja.**



Fuente: Manual de uso elaborado para SOBOCE S.A.

4) Ejecución del procedimiento (ver Imagen 18).

**Imagen 18: Ejecución del procedimiento A\_COPIAR\_DATOS.**



Fuente: Manual de uso elaborado para SOBOCE S.A.

#### 4.3.2. Transferencia de datos al formato del formulario

La prueba implica realizar la transferencia de los datos copiados al formato del formulario (ver Anexo 3) para esto solo es necesario la ejecución por parte del usuario. La Tabla 30 detalla los parámetros de la prueba.

**Tabla 30: Parámetros de la prueba de transferencia de datos**

Referencia	Valor
Responsable	Usuario (Analista de procesos)
Acción	Transferencia de datos
Datos de entrada	Datos de riesgos (Formato reporte).
Procedimiento	B_FORMATO_FORMULARIO
Datos de salida	Datos de riesgos (Formato formulario)

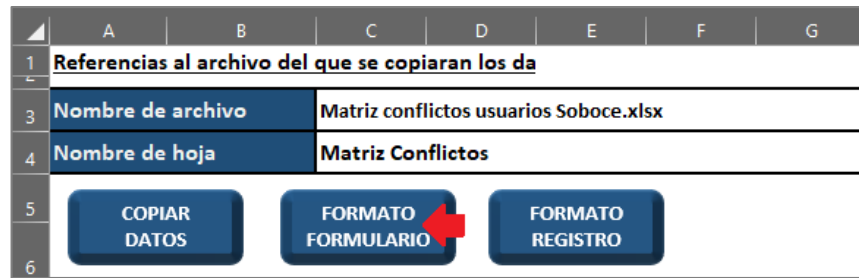
Fuente: Elaboración propia.

El procedimiento de la prueba es el siguiente:

1) En la hoja “COPIA”, ejecución del procedimiento (ver Imagen 19).



**Imagen 19: Ejecución del procedimiento B\_FORMATO\_FORMULARIO.**



Fuente: Manual de uso elaborado para SOBOCE S.A.

### 4.3.3. Creación de copia del formulario

La prueba implica realizar una copia del formulario solo con los datos necesarios y en formato sin macros (xlsx) para esto solo es necesario la ejecución por parte del usuario. La Tabla 31 detalla los parámetros de la prueba.

**Tabla 31: Parámetros de la prueba de creación de copia**

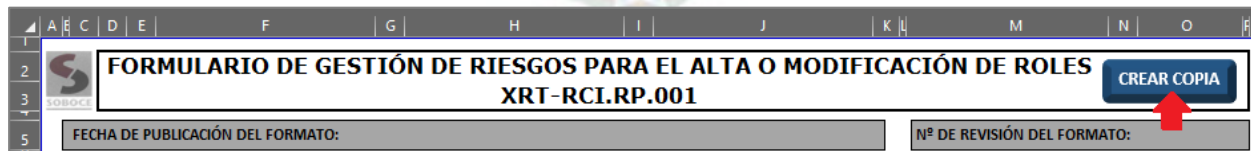
Referencia	Valor
Responsable	Usuario (Analista de procesos)
Acción	Creación de copia.
Datos de entrada	Ninguno
Procedimiento	A_COPIAR_DATOS
Datos de salida	Copia del formulario (extensión xlsx)

Fuente: Elaboración propia.

El procedimiento de la prueba es el siguiente:

- 1) En la hoja “FORMULARIO”, ejecución del procedimiento (ver Imagen 20)

**Imagen 20: Ejecución del procedimiento C\_CREAR\_COPIA.**



Fuente: Manual de uso elaborado para SOBOCE S.A.

### 4.3.4. Transferencia de datos al formato de registro

La prueba implica realizar la copia de datos desde la copia del formulario a la hoja con el formato de registro en el formulario automatizado, para esto es necesario el nombre del archivo

desde donde se realizará la copia y la ejecución por parte del usuario. La Tabla 32 detalla los parámetros de la prueba.

**Tabla 32: Parámetros de la prueba de transferencia de datos al formato de registro**

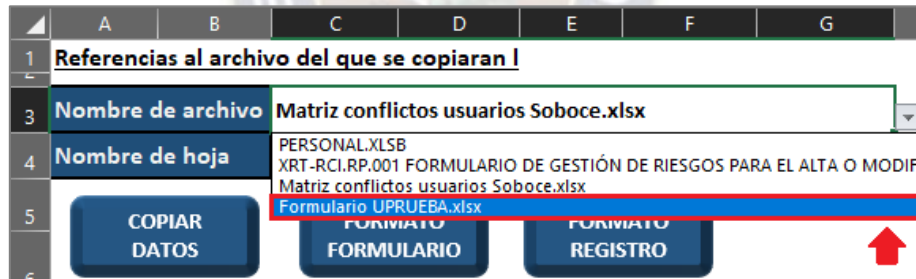
Referencia	Valor
Responsable	Usuario (Gestor de riesgos)
Acción	Transferencia de datos
Datos de entrada	Datos de riesgos
Procedimiento	D_FORMATO_REGISTRO
Datos de salida	Datos de riesgos

Fuente: Elaboración propia.

El procedimiento de la prueba es el siguiente:

- 1) En la hoja “COPIA”, selección del nombre del archivo (ver Imagen 21).

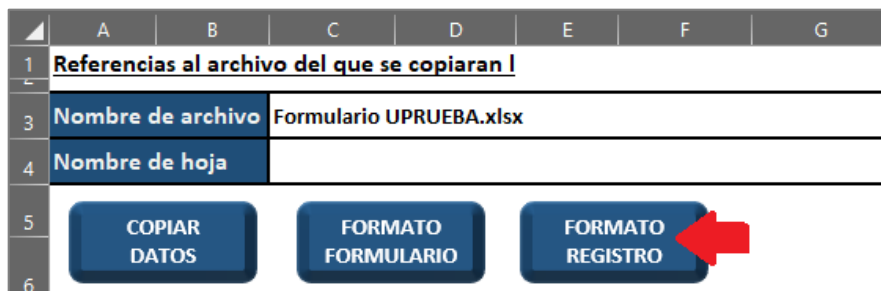
**Imagen 21: Selección nombre de formulario.**



Fuente: Manual de uso elaborado para SOBOCE S.A.

- 2) Ejecución del procedimiento (ver Imagen 22)

**Imagen 22: Ejecución del procedimiento D\_FORMATO\_REGISTRO.**



Fuente: Manual de uso elaborado para SOBOCE S.A.

#### 4.3.5. Registro en la Base de Datos

La prueba implica realizar una copia del formulario solo con los datos necesarios y en formato sin macros (xlsx) para esto solo es necesario la ejecución por parte del usuario. La Tabla 33 detalla los parámetros de la prueba.

**Tabla 33: Parámetros de la prueba de registro en base de datos**

Referencia	Valor
Responsable	Usuario (Analista de procesos)
Acción	Registro de datos en Base de Datos
Datos de entrada	Datos de riesgos y aprobaciones.
Procedimiento	A_COPIAR_DATOS
Datos de salida	Ninguno.

Fuente: Elaboración propia.

El procedimiento de la prueba es el siguiente:

- 1) En la hoja “REGISTRO”, ejecución del procedimiento (ver Imagen 23)

**Imagen 23: Ejecución del procedimiento E\_REGISTRO\_BD**

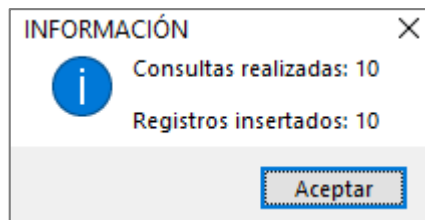


	A	B	C	D
22	REGISTRAR BD			
23	REFERENCIA1	USUARIO	AREA	MODULO
24	GG_001	UPRUEBA	COSTOS	SR
25	GG_001	UPRUEBA	COSTOS	SR
26	GG_001	UPRUEBA	COSTOS	SR
27	GG_001	UPRUEBA	COSTOS	SR
28	GG_001	UPRUEBA	COSTOS	SR

Fuente: Manual de uso elaborado para SOBOCE S.A.

- 2) Verificación (ver Imagen 24).

**Imagen 24: Mensaje informativo de registros insertados**



Fuente: Manual de uso elaborado para SOBOCE S.A.

#### 4.3.6. Modificación de registros

La prueba implica realizar una o varias modificaciones en la base de datos para finalizar los riesgos que se hayan eliminado o finalizar el riesgo debido a un cambio o baja del usuario que genera el riesgo. La Tabla 34 detalla los parámetros de la prueba.

**Tabla 34: Parámetros de la prueba de modificación de registros**

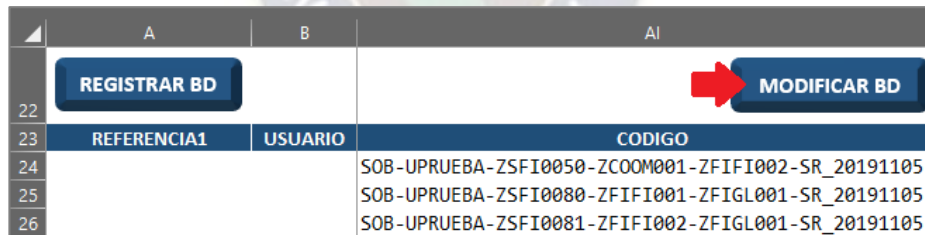
Referencia	Valor
Responsable	Usuario (Analista de procesos y Gestor de riesgos)
Acción	Finalización del riesgo.
Datos de entrada	Código de registro.
Procedimiento	F_MODIFICAR_BD
Datos de salida	Registro modificado.

Fuente: Elaboración propia.

El procedimiento de la prueba es el siguiente:

- 1) En la hoja “REGISTRO”, registro de códigos de identificación de registros.
- 2) Ejecución del procedimiento (ver Imagen 25)

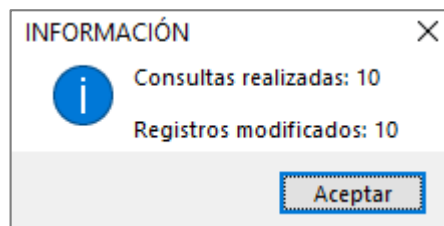
**Imagen 25: Ejecución del procedimiento F\_MODIFICAR\_BD**



Fuente: Fuente: Manual de uso elaborado para SOBOCE S.A.

- 3) Verificación (ver Imagen 26).

**Imagen 26: Mensaje informativo de registros modificados**



Fuente: Fuente: Manual de uso elaborado para SOBOCE S.A.

### 4.3.7. Generación de reporte

La prueba implica realizar varias consultas para generar distintos tipos de reporte desde la base de datos. La Tabla 35 detalla los parámetros de la prueba.

**Tabla 35: Parámetros de la prueba de generación de reporte**

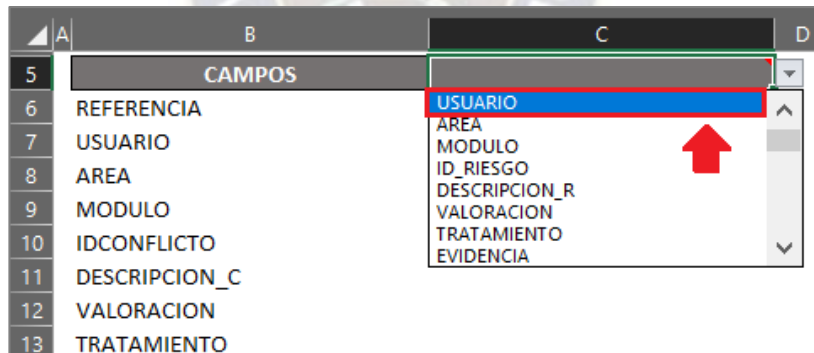
Referencia	Valor
Responsable	Usuario (Analista de procesos y Gestor de riesgos)
Acción	Generación de reporte
Datos de entrada	Consulta SQL
Procedimiento	G_REPORTE
Datos de salida	Registros

Fuente: Elaboración propia.

El procedimiento de la prueba es el siguiente:

- 1) En la hoja “**Consulta**”, Registro de los campos para el reporte.
- 2) Selección del campo de referencia (ver Imagen 27).

**Imagen 27: Selección de campo**



Fuente: Manual de uso elaborado para SOBOCE S.A.

- 3) Ejecución del procedimiento (ver Imagen 28).

**Imagen 28: Ejecución del procedimiento G\_REPORTE**

	A	B	C
1		<b>REPORTE</b>	
2		<b>CONSULTA DE REPORTE</b>	
3		Select ID-RIESGO, AREA, MODULO, TRATAMIENTO, FECHA_F From REGISTRO_NEW Where USUARIO IN ('UPRUEBA') Order By USUARIO;	
4			
5		<b>CAMPOS</b>	<b>USUARIO</b>
6		ID-RIESGO	UPRUEBA
7		AREA	
8		MODULO	
9		TRATAMIENTO	
10		FECHA_F	

Fuente: Manual de uso elaborado para SOBOCE S.A.





## Capítulo 5: Evaluación técnico-financiera

El software es esencial en cualquier área empresarial, ya que no solo es una unidad de soporte, sino que también es una fuente de innovación y mejoras para diversos procesos. Sin embargo, el desarrollo de software no es comparable con la producción de bienes industriales, dado que involucra una actividad mental y creativa. A pesar de que existen estándares para su desarrollo, en última instancia, todo depende de las personas y su nivel de especialización. Por esta razón, no es posible predecir el proceso de producción, ni estimar la calidad del producto final. Además, existe la problemática de la falta de uso extendido de metodologías para evaluar los proyectos de desarrollo de software (Aguilar, Ramos, & Andrade, 2017).

Existen diversas técnicas, modelos y filosofías utilizadas en la evaluación del desarrollo de software. Muchas de estas técnicas se desarrollaron en el contexto de programas sociales (software dedicado a la educación o la salud), por lo que abordan distintos valores, en lugar del enfoque más estrecho de los cálculos del valor actual neto. Entonces es importante considerar diferentes enfoques de evaluación en lugar de utilizar un único modelo rígido que no se adapte completamente al problema. Lo recomendable es combinar métodos contables con otros enfoques para obtener un valor significativo en el resultado final (Farbey & Finkelstein, 2001).

Realizar una evaluación basada en el VAN (Valor Actual Neto) requeriría una estimación de los beneficios futuros, lo cual no es recomendable, aunque sea posible, debido al grado de incertidumbre que presenta específicamente el caso de estudio. Motivo por el cual es preferible adoptar un enfoque conservador y basado en evidencia.

### 5.1. Comparación de procedimientos antes y después de la implementación

La Tabla 36 describe los procedimientos. Esta evaluación tiene como objetivo medir y analizar los cambios y mejoras obtenidos tras la implementación del software en un proceso o sistema, comparando los resultados y el rendimiento previo con los resultados y el rendimiento posterior a la implementación.

**Tabla 36: Comparación de procedimientos antes y después de la implementación**

<b>Actividad</b>	<b>Antes</b>	<b>Después</b>
Generación de la matriz de riesgos	Generación de reporte de riesgos de roles de usuario del sistema SAP	Generación de reporte de riesgos de roles de usuario del sistema SAP
Revisión y análisis de riesgos	(Analista de procesos) Revisión de un archivo Excel con datos históricos para identificar los riesgos que ya cuentan con un tratamiento o falsos positivos.	(Analista de procesos) Generación de reporte personalizado desde el formulario automatizado para identificar los riesgos que ya cuentan con un tratamiento o falsos positivos.
Registro de datos desde la matriz de riesgos	Copia manual y concatenación de los datos de transacción del sistema SAP en el formulario	Registro de datos de referencia para configurar la copia automática
Registro de datos complementarios	Registro manual de nombres de usuario, aprobadores y datos relacionados con el riesgo y su tratamiento.	Registro manual de nombres de usuario, aprobadores y datos relacionados con el riesgo y su tratamiento.
Protección de celdas	Protección manual de las celdas con la información de los riesgos	Creación de una copia del formulario y protección automática.
Envío de formulario	Envío del formulario vía e-mail al gestor de riesgos y usuarios clave.	Envío del formulario vía e-mail al gestor de riesgos y usuarios clave.
Análisis y establecimiento de tratamiento del riesgo	(Gestor de riesgos) Revisión de un archivo Excel con datos históricos para el análisis de riesgos similares.	(Gestor de riesgos) Generación de reporte personalizado desde el formulario automatizado para el análisis de riesgos similares.
Registro de los tratamientos	Registro manual de los tratamientos establecidos para los riesgos.	Registro manual de los tratamientos establecidos para los riesgos.

(Continúa en la siguiente página)

(Continuación)

<b>Actividad</b>	<b>Antes</b>	<b>Después</b>
Aprobación	Firmas del usuario clave del módulo SAP, Gerente del área correspondiente, Gestor de riesgos y Gerente general en caso de que se asuma un riesgo.	Firmas del usuario clave del módulo SAP, Gerente del área correspondiente, Gestor de riesgos y Gerente general en caso de que se asuma un riesgo.
Transferencia de datos a un formato de tabla	Transferencia de datos mediante funciones a un formato de tabla	Transferencia automática de los datos desde el formulario aprobado a un formato de tabla
Registro en base de datos	Copia manual de los datos para en un archivo Excel	Registro automático en una base de datos del servidor

Fuente: Elaboración Propia

## 5.2. Evaluación de la calidad del software

Muchas compañías optan por producir software "suficientemente bueno", reconociendo que puede contener errores conocidos. Su objetivo es satisfacer las necesidades de la mayoría de los usuarios y lanzarlas rápidamente al mercado, con planes de mejorar en versiones futuras. Aunque esta estrategia puede funcionar en algunos casos, existen argumentos en contra de esta práctica. Sin embargo, si una empresa logra capturar el mercado con la versión inicial, puede comprometerse a mejorar la calidad en versiones posteriores. (Pressman, 2010)

Ante esta situación se han desarrollado varios estándares para evaluar la calidad del software. La norma ISO 9126 establece el estándar para identificar las características clave del software. Estas características no necesariamente conducen a una medición directa. Sin embargo, proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema (International Organization for Standardization [ISO], 2001). En esta norma se identifican seis características clave de calidad:

- 1) **Funcionalidad:** atributos que permiten calificar si el software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.
- 2) **Confiabilidad:** atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.
- 3) **Usabilidad:** atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.
- 4) **Eficiencia:** atributos que permiten evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados.

- 5) **Mantenibilidad:** atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea por la corrección de errores o por el incremento de funcionalidades.
- 6) **Portabilidad:** atributos que permiten evaluar si el software tiene la capacidad de ser transferido de un ambiente a otro.

En la Tabla 37 se detalla el porcentaje de cumplimiento de los atributos utilizados (ver Anexo 5: Atributos de evaluación de la calidad del software, pág. 102) para evaluar el cumplimiento de las características de calidad del software. Según estos resultados, el software cumple satisfactoriamente con las características de funcionalidad, usabilidad y eficiencia. Las características por mejorar son la confiabilidad y la mantenibilidad. Debido a que el formulario es un complemento y no un software en sí mismo, presenta deficiencias con la característica de portabilidad. Con base en estos resultados y considerando las características del formulario automatizado se puede afirmar que es una herramienta que cumple satisfactoriamente con los estándares de un software de calidad según la norma ISO 9126.

**Tabla 37: Cumplimiento de las características de calidad.**

<b>Característica</b>	<b>Cumplimiento</b>
Funcionalidad	100%
Confiabilidad	66%
Usabilidad	100%
Eficiencia	100%
Mantenibilidad	75%
Portabilidad	50%

Fuente: Elaboración propia con base en la norma ISO 9126.

### **5.3. Análisis costo-beneficio**

El análisis de costo-beneficio se utiliza para identificar, medir y comparar los beneficios y costos de un proyecto o programa de inversión. Se aplica tanto a proyectos públicos como privados con el objetivo de determinar si representan un uso eficiente de los recursos. Cuando una empresa privada o pública evalúa un proyecto desde una perspectiva puramente comercial, solo se consideran los beneficios y costos que afectan la rentabilidad de la empresa (Campbell & Brown, 2003).

- **Inversión:**

El proyecto surge como una propuesta del autor mientras cumplía sus funciones como analista de procesos con la tarea de elaborar un formulario para el proceso de gestión de riesgos de usuarios en el sistema SAP. Dado que la automatización no era un requisito específico de la empresa y tampoco era parte de las funciones para las cuales el autor fue contratado, no se asignó un presupuesto inicial como inversión del proyecto.

- **Costos del proyecto:**

El proyecto fue desarrollado en 3 fases con un total de Bs 4066, en la Tabla 38 se detalla los costos de cada fase.

**Tabla 38: Costos del proyecto**

<b>Fase</b>	<b>Descripción</b>	<b>Costo [Bs]</b>
Fase 1	Diseño del formulario para 10 usuarios, 10 riesgos y 2 funciones para cada riesgo. Al ser una propuesta no se considera un costo.	0
Fase 2	Rediseño del formulario para 20 usuarios, 20 riesgos, 3 funciones para cada riesgo y refactorización del código para mejorar la eficiencia. (El costo incluye documentación y capacitación)	1700
Fase 3	Inclusión de nuevos valores para las listas desplegables del formulario y campos condicionados. Inclusión de funcionalidades de confirmación para el registro o modificación en la base de datos y la generación de reportes. (El costo incluye documentación y capacitación)	2366

Fuente: Elaboración propia

- **Beneficios del proyecto**

El proyecto no tiene impacto directo sobre los ingresos de SOBOCE S.A., ya que se implementó en un área administrativa, por lo que el principal beneficio económico es la reducción de costos al eliminar tareas manuales en el proceso para el cual fue diseñado el formulario automatizado.

El costo por minuto del personal (promedio del sueldo de 6 personas) encargado del registro del formulario es:

$$\text{Costo del personal} = 9000 \frac{\text{Bs}}{\text{mes}} \times \frac{1 \text{ mes}}{160 \text{ h}} \times \frac{1 \text{ h}}{60 \text{ min}} = 0,94 \frac{\text{Bs}}{\text{min}}$$

Esto significa que cada minuto que dedica una persona al registro del formulario tiene un costo de Bs 0,94.



El beneficio total por la reducción de tiempo en el registro de 327 riesgos en tres años (2019 a 2022) se calcula considerando que cada riesgo se procesa en 10 minutos aproximadamente (ver Imagen 7: Diagrama de flujo del proceso sin la automatización) y que cada minuto del personal tiene un costo de Bs 0,94:

$$\text{Beneficio} = 327 \text{ riesgos} \times \frac{10 \text{ min}}{1 \text{ riesgo}} \times \frac{0,94 \text{ Bs}}{1 \text{ min}} = 3073,80 \text{ Bs}$$

Entonces, al reducir el tiempo dedicado al registro de riesgos en el formulario, se obtiene un beneficio económico de Bs 3,073.80. Este beneficio representa el ahorro en costos de personal al automatizar el registro en el formulario.

- **Relación costo-beneficio**

Con base en los cálculos realizados, la relación costo-beneficio del proyecto es:

$$RCB = \frac{3073,80}{4066} = 0,75$$

Los beneficios del proyecto son inferiores a los costos asociados. Pero se debe tener en cuenta, que en lo referente a software, la recuperación de la inversión se basa en los beneficios tangibles e intangibles que aporta a la organización. Además, la inversión en software generalmente se recupera a largo plazo considerando las altas inversiones que se realizan en el desarrollo de software personalizado o la adquisición de licencias, por ejemplo SAP o Microsoft.

#### 5.4. Evaluación de impacto

La Tabla 39 presenta una serie de métricas clave para evaluar el impacto de la implementación del formulario automatizado. Al comparar los resultados antes y después de la implementación, se puede determinar el grado de mejora logrado mediante la automatización.

**Tabla 39: Métricas de evaluación**

<b>Métrica</b>	<b>Antes</b>	<b>Después</b>
Tiempo promedio de completado (s)	600	30
Cantidad de riesgos trabajados	5	327
Errores de entrada de datos (promedio por formulario)	3	0
Ahorro de recursos (Bs)	0	3073,80
Generación de informes	0	1

Fuente: Elaboración propia



La principal mejora se refleja en la reducción del tiempo empleado para el registro del formulario de un tiempo promedio de 10 minutos que variaba por el número de riesgos a 30 segundos independientemente del número de riesgos registrados en el formulario. Esto contribuye a la productividad ya que puede realizarse un mayor número de registros en menor tiempo. Además la implementación reduce los errores de registro a cero y proporciona un reporte personalizado que facilita la revisión general del formulario.



## Capítulo 6: Conclusiones y recomendaciones

### 6.1. Conclusiones

Se logró cumplir el objetivo principal de automatizar el formulario del procedimiento de gestión de riesgos para el alta o modificación de roles de usuarios del sistema SAP en SOBOCE S.A., La implementación del formulario automatizado ha permitido una mayor eficiencia y agilidad en el proceso contribuyendo a mejorar la productividad y eficacia del área de soluciones y proyectos.

Además del objetivo general, también se puede afirmar que se han cumplido los objetivos específicos de acuerdo con el siguiente detalle:

- 1) En el marco teórico y aplicativo se logró sistematizar la información secundaria relacionada con la automatización de Microsoft Excel, lo que contribuirá a optimizar el manejo de datos y tareas repetitivas en la organización.
- 2) Se realizó un diagnóstico del proceso de gestión de riesgos para el alta o modificación de roles de usuarios del sistema SAP en SOBOCE S.A., lo que permitió identificar las actividades que podían mejorarse y plantear los requerimientos para automatizar el formulario.
- 3) Se desarrolló un formulario automatizado basado en los requerimientos del diagnóstico, obteniendo una solución eficiente y personalizada para SOBOCE S.A. Además, la implementación del código generado ha demostrado ser efectiva y segura en un entorno real y al más alto nivel empresarial, lo que lo convierte en una opción atractiva para su uso en diferentes procesos.
- 4) Se evaluó el formulario automatizado desde el aspecto financiero mediante el análisis de costo-beneficio y desde el aspecto técnico mediante el estándar de la norma ISO 9126. En 3 años de uso la empresa ha recuperado 75% de su inversión, lo cual es aceptable considerando que el retorno de la inversión de software es a largo plazo. Además, de acuerdo con la norma ISO 9126, el software cumple satisfactoriamente con las características de calidad, por lo que se puede afirmar que la evaluación es positiva.

## 6.2. Recomendaciones

El presente proyecto ha logrado desarrollar una herramienta de software para automatizar las tareas repetitivas en un proceso de una empresa de alto nivel como SOBOCE S.A. Sin embargo, existe dependencia del desarrollador, por lo que se recomienda generar un plan de continuidad que permita el correcto funcionamiento y la posible mejora del formulario automatizado.

Adicionalmente, se tienen las siguientes recomendaciones:

- 1) Realizar un seguimiento y evaluación continuos del desempeño del formulario automatizado para garantizar su eficacia y eficiencia. Esto se puede lograr a través de una retroalimentación constante por parte de los usuarios.
- 2) Aprovechar el conocimiento adquirido sobre el funcionamiento y la manipulación del código del formulario automatizado para continuar con la automatización de otros procesos que utilicen la aplicación de Microsoft Excel. El desarrollo exitoso de estas herramientas de software dependerá en gran medida del análisis, la definición de requisitos, y el diseño de la arquitectura.
- 3) Complementar el plan de formación y capacitación existente en la empresa, con temas relacionados con el desarrollo de complementos para las aplicaciones de Microsoft Office. Este conocimiento permitirá al personal crear herramientas personalizadas que agilicen los procesos, fomenten la estandarización de las actividades e incluso posibiliten el diseño de prototipos para el desarrollo de software más complejo.
- 4) Mejorar la característica de confiabilidad del formulario automatizado mediante una funcionalidad que registre los errores ocurridos en una base de datos para medir la frecuencia de fallas y tomar acciones correctivas sobre la base de los datos.
- 5) Desarrollar la solución del formulario automatizado como un complemento instalable en la plataforma de Visual Studio para cumplir con la característica de portabilidad de la norma ISO 9126.

## Área III – Análisis de la actividad laboral

La experiencia laboral en desarrollo de software, análisis de procesos y manejo de datos ha dotado al autor de habilidades fundamentales para resolver problemas de manera efectiva. Entre estas habilidades se encuentran la capacidad de descomponer problemas complejos en tareas más pequeñas, aplicar el pensamiento lógico en el diseño de algoritmos y soluciones, así como la habilidad de abstraer problemas para encontrar soluciones generales.

Los conocimientos requeridos para el desempeño de estas actividades se enumeran a continuación:

1. Programación y diseño de software para desarrollar aplicaciones escalables. Esto implica tener conocimientos de patrones de diseño, sintaxis de lenguajes de programación (VBA, VB.NET y Python), buenas prácticas de codificación, modularidad y mantenibilidad del software.
2. Conceptos y principios de las bases de datos para realizar consultas y la gestión eficiente de grandes conjuntos de datos.
3. Lenguaje de consulta SQL para insertar, modificar y extraer datos desde una base de datos.
4. Técnicas y métodos de análisis de datos para descubrir patrones, tendencias, correlaciones y generar información útil a partir de los conjuntos de datos disponibles.
5. Conocimiento en sistemas para diseñar aplicaciones eficientes, comprendiendo las interacciones entre los componentes del sistema.
6. Metodología para comprender a detalle el funcionamiento de procesos con el objetivo de mejorar su eficiencia.

Además de las habilidades y conocimientos en el manejo de datos, el análisis de procesos y el desarrollo de software, también deben enfrentarse los siguientes desafíos:

1. Protección de los datos e información de una empresa contra el acceso no autorizado y las filtraciones.
2. Uso indebido de datos recolectados en la empresa.
3. Evitar la introducción de sesgos en los datos para beneficio propio o de terceros.
4. Promover la transparencia de los procesos y resultados para que los usuarios puedan comprender las acciones realizadas.

Es esencial abordar estos desafíos éticos mediante el conocimiento de la normativa de cada empresa u organización referente al manejo de datos, la educación y conciencia sobre el uso responsable de datos. La ética debe estar presente en toda actividad laboral porque esto garantiza la formación de una relación de confianza entre la empresa y el profesional.

## Bibliografía

- Aguilar, H. M., Ramos, L. A., & Andrade, R. A. (2017). Modelo de evaluación de proyectos de software utilizando una lógica difusa compensatoria. *NovaRua*, 8(14), 90-104.
- Amelot, M. (2016). *VBA Excel 2016: Programación en Excel macros y lenguaje VBA*. Ediciones ENI.
- Bel, W. (2020). *Algoritmos y estructuras de datos en Python: un enfoque ágil y estructurado*. Uader.
- Campbell, H. F., & Brown, R. P. (2003). *Benefit-cost analysis: financial and economic appraisal using spreadsheets*. Cambridge University Press.
- Cervantes, H., & Kazman, R. (2016). *Designing software architectures : a practical approach* . Pearson.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms*. MIT press.
- Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. (2013). *Fundamentals of business process management*. Springer.
- Farbey, B., & Finkelstein, A. (2001). *Evaluation in Software Engineering: ROI, but more than ROI*. Obtenido de City Research Online: <https://openaccess.city.ac.uk/id/eprint/26482/>
- Franklin Fincowsky, E. B. (2009). *Organización de empresas*. McGraw Hill.
- Gomaa, H. (2011). *Software modeling and design: UML, use cases, patterns, and software architectures*. Cambridge University Press.
- IBM. (1 de Mayo de 2023). *¿Qué es la automatización?* Obtenido de <https://www.ibm.com/es-es/topics/automation>
- International Organization for Standardization [ISO]. (1985). ISO 5807:1985 Information processing - Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.
- International Organization for Standardization [ISO]. (2001). ISO/IEC 9126:2001 Software engineering - Product quality - Part 1: Quality model.
- International Organization for Standardization [ISO]. (2015). ISO 9000:2015 Quality management systems - Fundamentals and vocabulary.
- International Organization for Standardization [ISO]. (2015). ISO 9001:2015 Quality management systems - Requirements.

- Joyanes Aguilar, L. (2008). *Fundamentos de programación: Algoritmos, estructura de datos y objetos*. McGraw-Hill.
- Kendall, K. E., & Kendall, J. E. (2011). *Análisis y diseño de sistemas*. Pearson.
- Knuth, D. (1997). *The Art of Computer Programming: Fundamental Algorithms, volume 1*. Addison Wesley.
- Laudon, K., & Laudon, J. (2012). *Sistemas de información gerencial*. México: Pearson.
- Linoff, G. S. (2015). *Data Analysis Using SQL and Excel*. John Wiley & Sons.
- Martin, R. C. (2009). *Clean code: a handbook of agile software craftsmanship*. Pearson.
- Microsoft. (2021). *Crear un diagrama de flujo de funciones cruzadas*. Obtenido de <https://support.microsoft.com/es-es/office/crear-un-diagrama-de-flujo-de-funciones-cruzadas-4a403033-9787-454f-b87e-b88452c47a21>
- Microsoft. (1 de Mayo de 2023). *Introducción al desarrollo de soluciones de Office (VSTO)*. Obtenido de <https://learn.microsoft.com/es-es/visualstudio/vsto/office-solutions-development-overview-vsto?view=vs-2022>
- Niebel, B., & Freivalds, A. (2009). *Ingeniería industrial: Métodos, estándares y diseño del trabajo*. McGraw-Hill.
- O'brien, J. A., & Marakas, G. M. (2006). *Sistemas de información gerencial*. McGraw-Hill.
- Oppel, A., & Sheldon, R. (2009). *Fundamentos de SQL*. McGraw-Hill.
- Pantigoso, R. (2006). *Algoritmos y Diagramas de flujo aplicados en C++*. Grupo Editorial Megabyte.
- Pressman, R. S. (2010). *Ingeniería del Software. Un enfoque práctico*. McGraw-Hill.
- SAP. (2023). *SAP Help Portal (Documentation)*. Obtenido de [https://help.sap.com/docs/SAP\\_ACCESS\\_CONTROL?locale=en-US](https://help.sap.com/docs/SAP_ACCESS_CONTROL?locale=en-US)
- SAP. (2023). *What is SAP HANA?* Obtenido de <https://www.sap.com/products/technology-platform/hana/what-is-sap-hana.html>
- SAP. (2023). *What is SAP?* Obtenido de <https://www.sap.com/about/what-is-sap.html>
- SOBOCE S.A. (2018). *Gestión de procesos e información documentada*. La Paz, Bolivia.
- Von Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. New York: George Braziller.
- Wasson, C. S. (2006). *System Analysis, Design and Development: Concepts, Principles, and Practices*. John Wiley & Sons.



## Anexo 1: Sintaxis del lenguaje VBA

Tabla 40: Sintaxis del lenguaje VBA

Sintaxis	Descripción de elementos
<pre>Dim nombre [As tipo]</pre>	<p><b>nombre:</b> debe seguir las convenciones estándar para nombrar variables.</p>
	<p><b>[tipo]:</b> tipo de dato de la variable (Boolean, Integer, String, Variant)</p>
<pre>If condición Then   [instrucciones] Else   [instrucciones_else] End If</pre>	<p><b>Condición:</b> Expresión numérica o de cadena que puede ser evaluada como Verdadero (True) o Falso (False).</p>
	<p><b>[instrucciones]:</b> Una o más instrucciones que se ejecutan si la condición asociada es verdadera (True).</p>
	<p><b>[instrucciones_else]:</b> Una o más instrucciones ejecutadas si la condición asociada no es Verdadero (True).</p>
<pre>Select Case expresión_prueba   Case expresión_1     instrucción_1   [Case expresión_2     [instrucción_2]]   ...   [Case expresión_n     [instrucción_n]] End Select</pre>	<p><b>expresión_prueba:</b> Cualquier expresión numérica o de cadena.</p>
	<p><b>expresión_n:</b> Cualquier expresión numérica o expresión de cadena con el que se compara expresión_prueba.</p>
	<p><b>instrucción_n:</b> Una o más instrucciones ejecutadas si expresión_prueba coincide con cualquier expresión_n.</p>
<pre>Do [While condición]   [instrucciones]   [Exit Do]   [instrucciones] Loop</pre>	<p><b>condición:</b> Expresión numérica o expresión de cadena que es Verdadero (True) o Falso (False). Si no se especifica, condición se considera como Falso (False).</p>
	<p><b>Instrucciones:</b> Una o más instrucciones que se repiten mientras o hasta que condición sea Verdadero (True).</p>
	<p><b>[Exit Do]:</b> Salida de la instrucción Do. Se transfiere el control a la sentencia que sigue a Loop.</p>

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de: <https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

Sintaxis	Descripción de elementos
<pre>Set variable_objeto = {[New] expresión_objeto   Nothing}</pre>	<p><b>variable_objeto:</b> Nombre de la variable o de la propiedad; sigue las convenciones estándar de nombres de variables.</p> <p><b>New:</b> Se utiliza normalmente durante una declaración para permitir la creación implícita de un objeto.</p> <p><b>expresión_objeto:</b> Expresión que consiste en el nombre de un objeto (variable, función o método que devuelve un objeto).</p> <p><b>Nothing:</b> Interrumpe una asociación de variable_objeto con cualquier objeto específico. Se liberan todos los recursos del sistema y de memoria asociados con el objeto.</p>
<pre>For contador = principio To fin   [instrucciones]   [Exit For]   [instrucciones] Next [contador]</pre>	<p><b>contador:</b> Variable numérica que se utiliza como contador de bucle.</p> <p><b>principio:</b> Valor inicial del contador.</p> <p><b>fin:</b> Valor final del contador.</p> <p><b>instrucciones:</b> Una o más instrucciones entre For y Next que se ejecutan un número especificado de veces.</p> <p><b>Exit For:</b> Salida de la instrucción For. Se transfiere el control a la sentencia que sigue a Next.</p>
<pre>For Each elemento In grupo   [instrucciones]   [Exit For]   [instrucciones] Next [elemento]</pre>	<p><b>elemento:</b> Variable que se utiliza para iterar por los elementos de grupo (conjunto de objetos o matriz).</p> <p><b>grupo:</b> Nombre de un conjunto de objetos o de una matriz (excepto una matriz de tipos definidos por el usuario).</p> <p><b>instrucciones:</b> Una o más instrucciones que se ejecutan para cada elemento de un grupo.</p> <p><b>Exit For:</b> Salida de la instrucción For Each. Se transfiere el control a la sentencia que sigue a Next.</p>

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de: <https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

**Tabla 41: Operadores**

<b>Operador</b>	<b>Descripción</b>
=	Asigna un valor a una variable. También se utiliza para verificar si dos variables o valores son iguales.
< >	Verifica si dos variables o valores son distintas.
AND	Si todas las expresiones tienen el valor Verdadero (True), el resultado es Verdadero (True). Si una de las expresiones tiene el valor Falso (False), el resultado es Falso (False).
OR	Si por lo menos una de las expresiones tiene el valor Verdadero (True), el resultado es Verdadero (True).
NOT	Devuelve el valor contrario de la expresión.
&	Fuerza la concatenación de dos o más expresiones (String, Integer).

Fuente: Elaboración propia con base en la documentación de Microsoft obtenida de: <https://learn.microsoft.com/es-es/office/vba/library-reference/concepts/getting-started-with-vba-in-office>

## Anexo 2: Formulario

**Tabla 42: Datos del formulario**

Sección	Dato	Descripción
Datos generales	Número de Ticket	Número asignado por Service Desk (Atención al cliente interno de SOBOCE)
	Referencia 1	Código alfanumérico generado por el reporte.
	Referencia 2	Código alfanumérico generado por el Funcional de soluciones y Proyectos.
	Empresa	SOBOCE, ESMICAL o YIBSA
	Fecha	Fecha de solicitud.
	Solicitante	Líder de módulo en el que se identifican riesgos en los roles de usuario.
	Cargo	Cargo del líder de módulo.
	Área	Nombre del área al que pertenece el solicitante (Líder de Módulo).
	Modulo	Módulo SAP en el que se identifican riesgos en los roles de usuario.
Usuarios	Usuario	Código del usuario cuyo rol presenta el riesgo (El formulario acepta a 20 usuarios). Los usuarios deben ser del mismo modulo.
	Observaciones	Comentario general respecto al formulario.
Riesgos	ID-Riesgo	Código alfanumérico del riesgo.
	Descripción Riesgo	Explicación del riesgo.
	Valoración	Importancia que se le concede al riesgo (Alto, Medio, Bajo).
	Tratamiento	Acción ante el riesgo (Asumir, Controlar, Falso positivo).
	Evidencia	Medio por el que se prueba la realización del control (Archivo File Server, Correo, Informe manual).
	Periodicidad	Periodo de tiempo para presentar la evidencia (Semanal, Mensual, Anual).
	Vigencia	Fecha límite del control.
	Responsable	Cargo del responsable de realizar el control.
	Función	Código alfanumérico de la función que causa el riesgo (Cada riesgo puede tener 2 o 3 funciones).
	Descripción Función	Descripción de la función que causa el riesgo.
	Transacciones	Transacción(es) de la función que causa el riesgo.
Aprobadores	Usuario Clave	Nombre del líder de módulo.
	Cargo Usuario Clave	Cargo del líder de módulo.
	Nombre Aprobador	Nombre del Gerente o Subgerente correspondiente del módulo.
	Cargo Aprobador	Cargo del Gerente o Subgerente correspondiente del módulo.
	Gestor de riesgos	Nombre del responsable de gestionar y coordinar el tratamiento de los riesgos
	Gerente	Nombre del Gerente General de SOBOCE en caso de que no se posible mitigar o controlar el riesgo y sea asumido.



**FORMULARIO DE GESTIÓN DE RIESGOS PARA EL ALTA O MODIFICACIÓN DE ROLES**  
XRT-RCI.RP.001

CREAR COPIA

FECHA DE PUBLICACIÓN DEL FORMATO:		Nº DE REVISIÓN DEL FORMATO:	
Número de Ticket: <small>(Llenado solo por Service Desk)</small>	<b>REGISTRAR NÚMERO DE TICKET</b>	Referencia 1	
		Referencia 2	
EMPRESA:		Fecha:	
Nombre Solicitante (Líder de Módulo)		Cargo	
Módulo		Área	

**1. USUARIO(S)**

1)		6)		11)		16)		Observaciones:
2)		7)		12)		17)		
3)		8)		13)		18)		
4)		9)		14)		19)		
5)		10)		15)		20)		

**2. MITIGACIÓN DE RIESGOS**

1) ID-Riesgo: Valoración: Tratamiento: Tipo de evidencia: Periodicidad: Vigencia hasta:

Descripción Riesgo: Control de Mitigación: Cargo Responsable:

Función 1: Descripción función 1: Transacciones función 1:

Función 2: Descripción función 2: Transacciones función 2:

Función 3: Descripción función 3: Transacciones función 3:

2) ID-Riesgo: Valoración: Tratamiento: Tipo de evidencia: Periodicidad: Vigencia hasta:

Descripción Riesgo: Control de Mitigación: Cargo Responsable:

Función 1: Descripción función 1: Transacciones función 1:

Función 2: Descripción función 2: Transacciones función 2:

Función 3: Descripción función 3: Transacciones función 3:

3) ID-Riesgo: Valoración: Tratamiento: Tipo de evidencia: Periodicidad: Vigencia hasta:

Descripción Riesgo: Control de Mitigación: Cargo Responsable:

Función 1: Descripción función 1: Transacciones función 1:

Función 2: Descripción función 2: Transacciones función 2:

Función 3: Descripción función 3: Transacciones función 3:

**3. APROBACIONES**

USUARIO CLAVE	GERENCIA NACIONAL RESPONSABLE	GESTOR DE RIESGOS SOBOCE
Fecha:	Fecha:	Fecha:
	GERENTE GENERAL SOBOCE	
	Fecha:	

Como Aprobador, declaro tener conocimiento del riesgo y el método de control para la mitigación del mismo.

### Anexo 3: Diagramas de flujo

Imagen 29: Diagrama de flujo del algoritmo de copia de datos

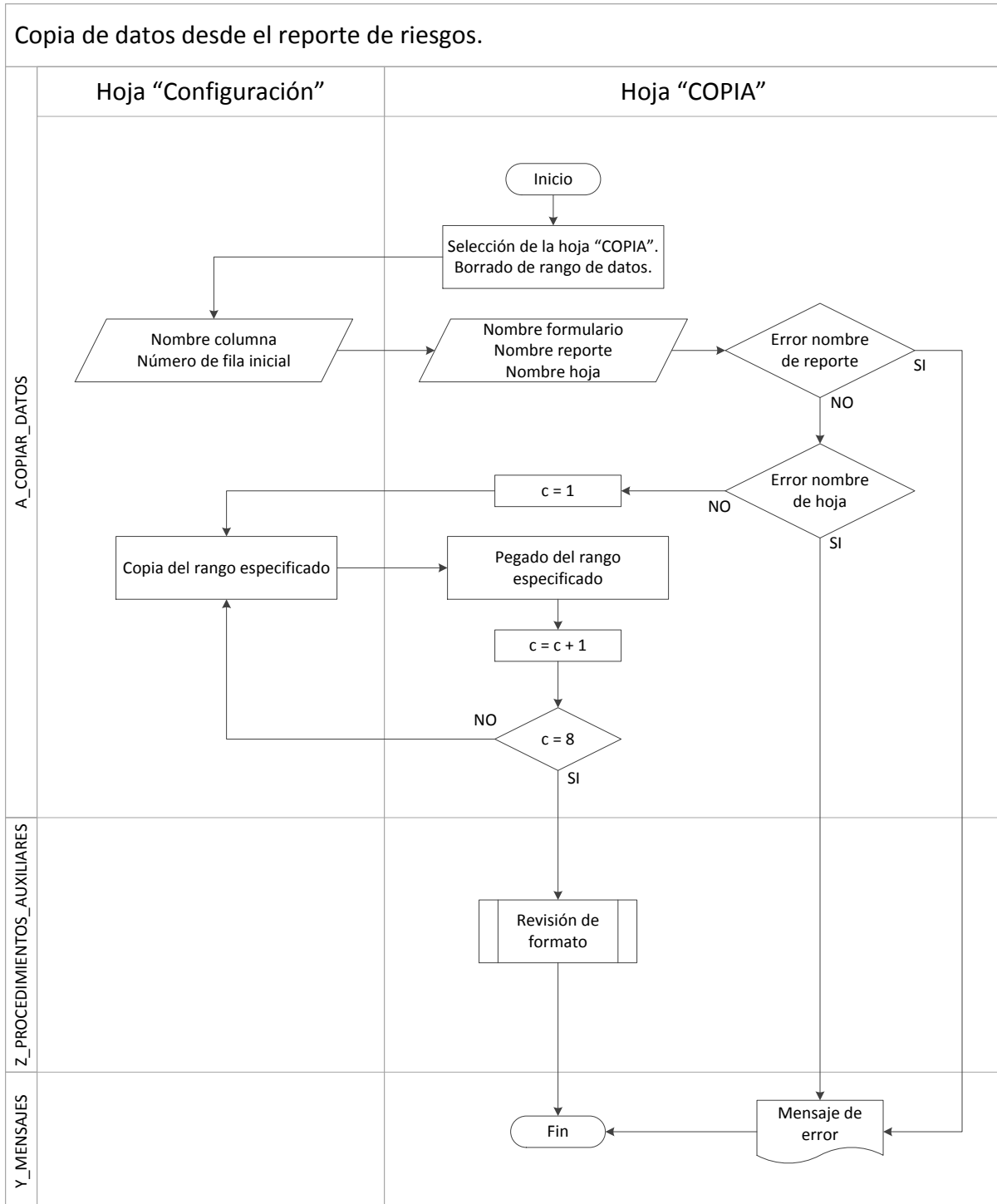




Imagen 30: Diagrama de flujo del algoritmo de revisión

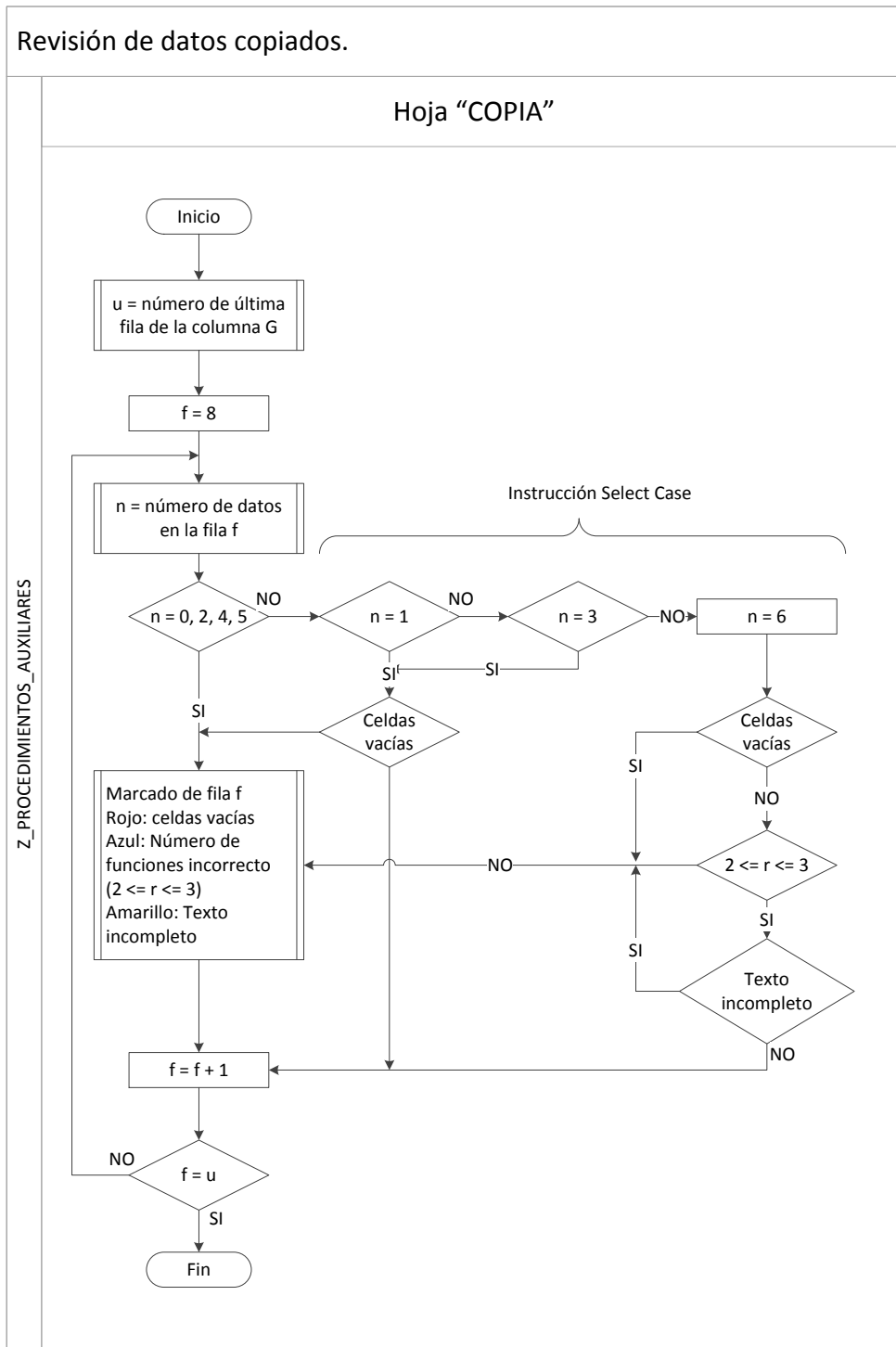
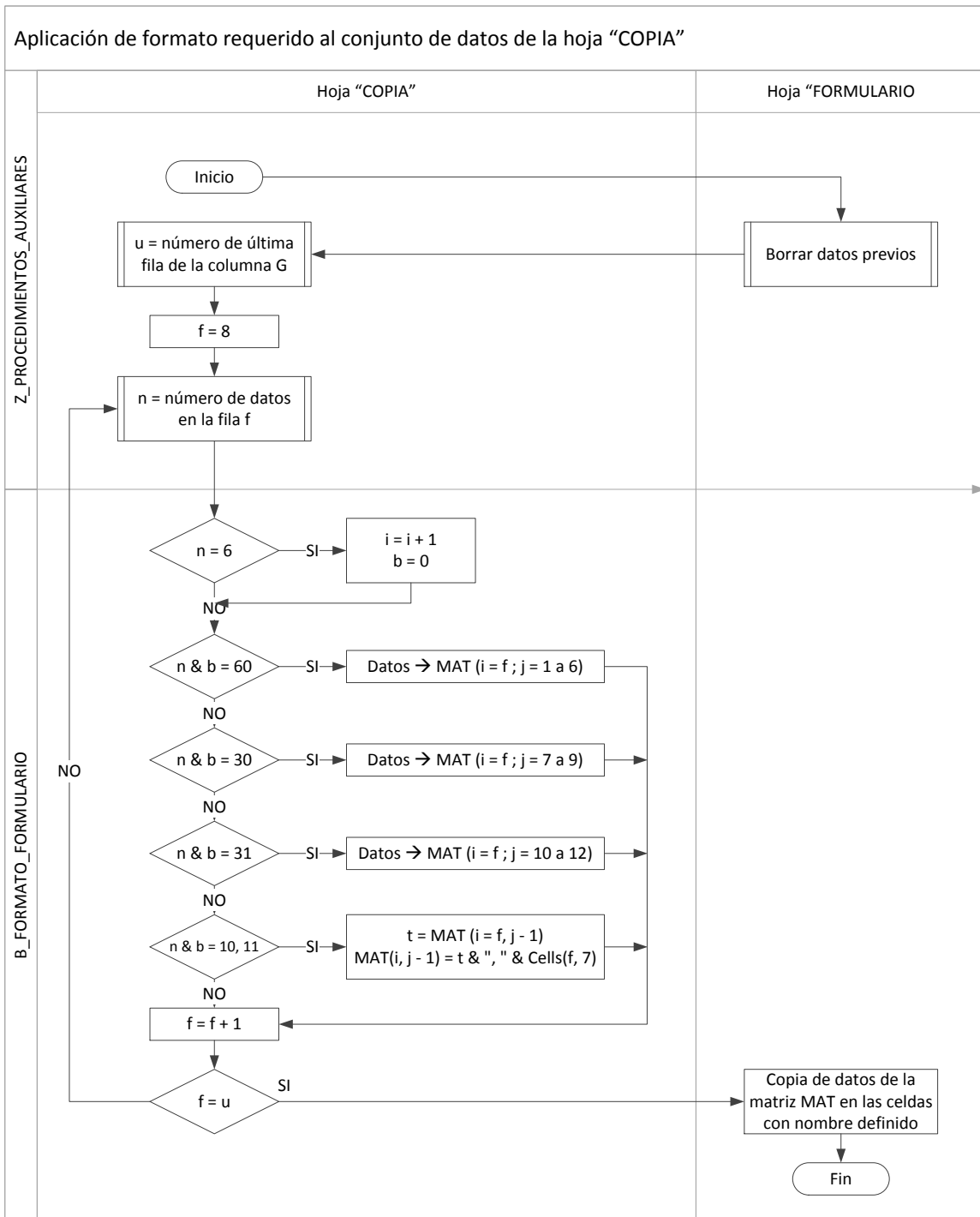
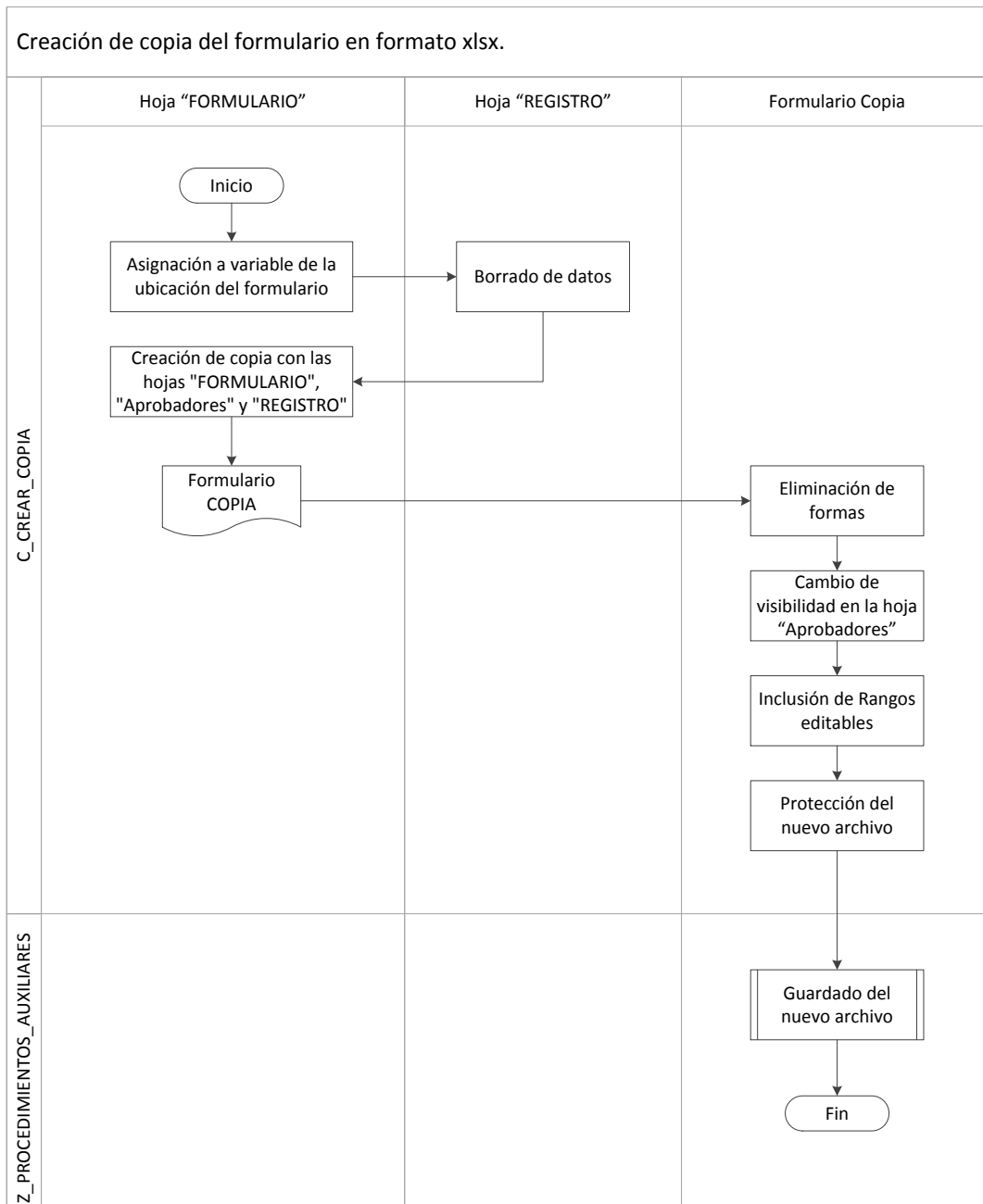


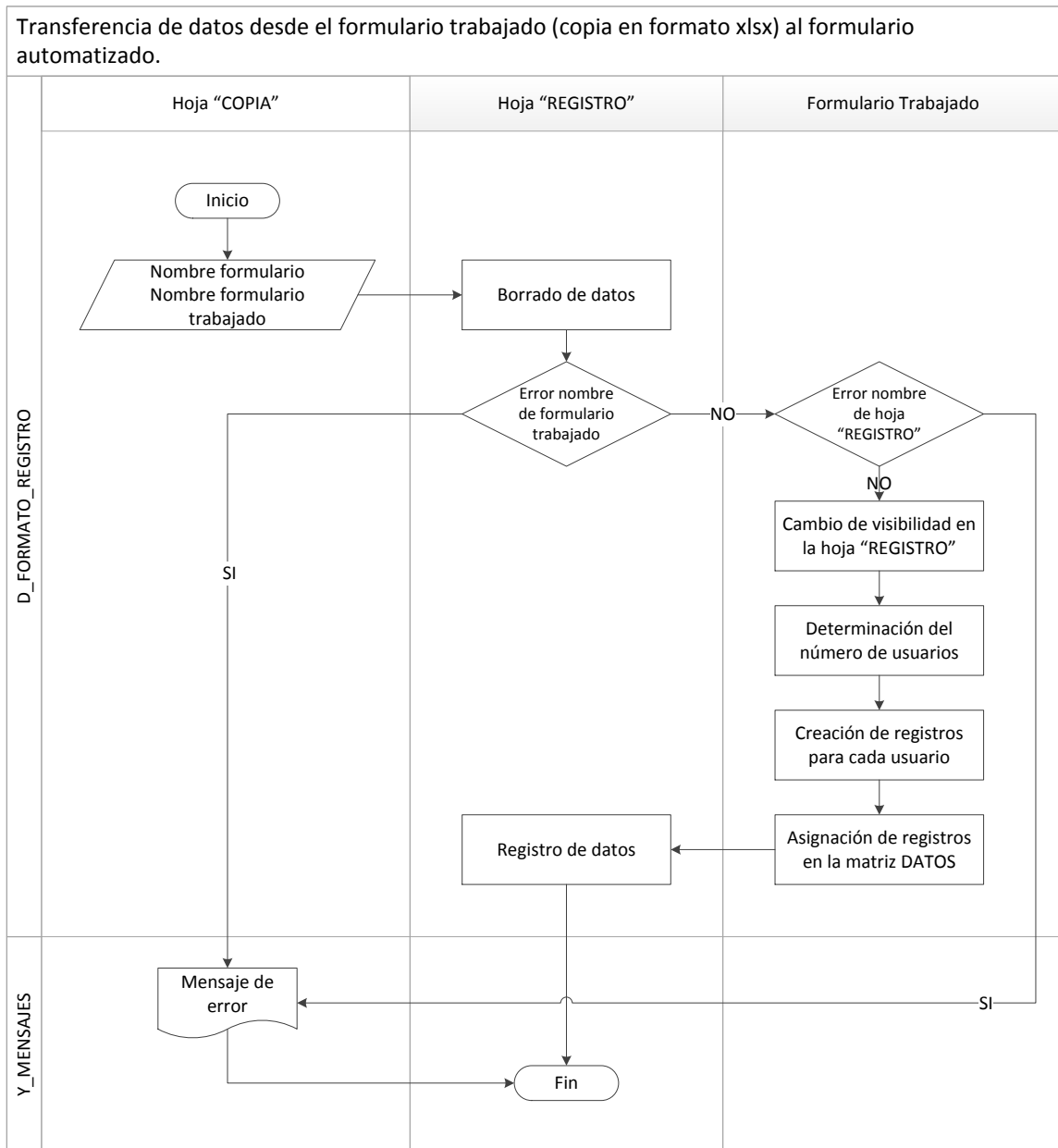
Imagen 31: Diagrama de flujo del algoritmo de formato



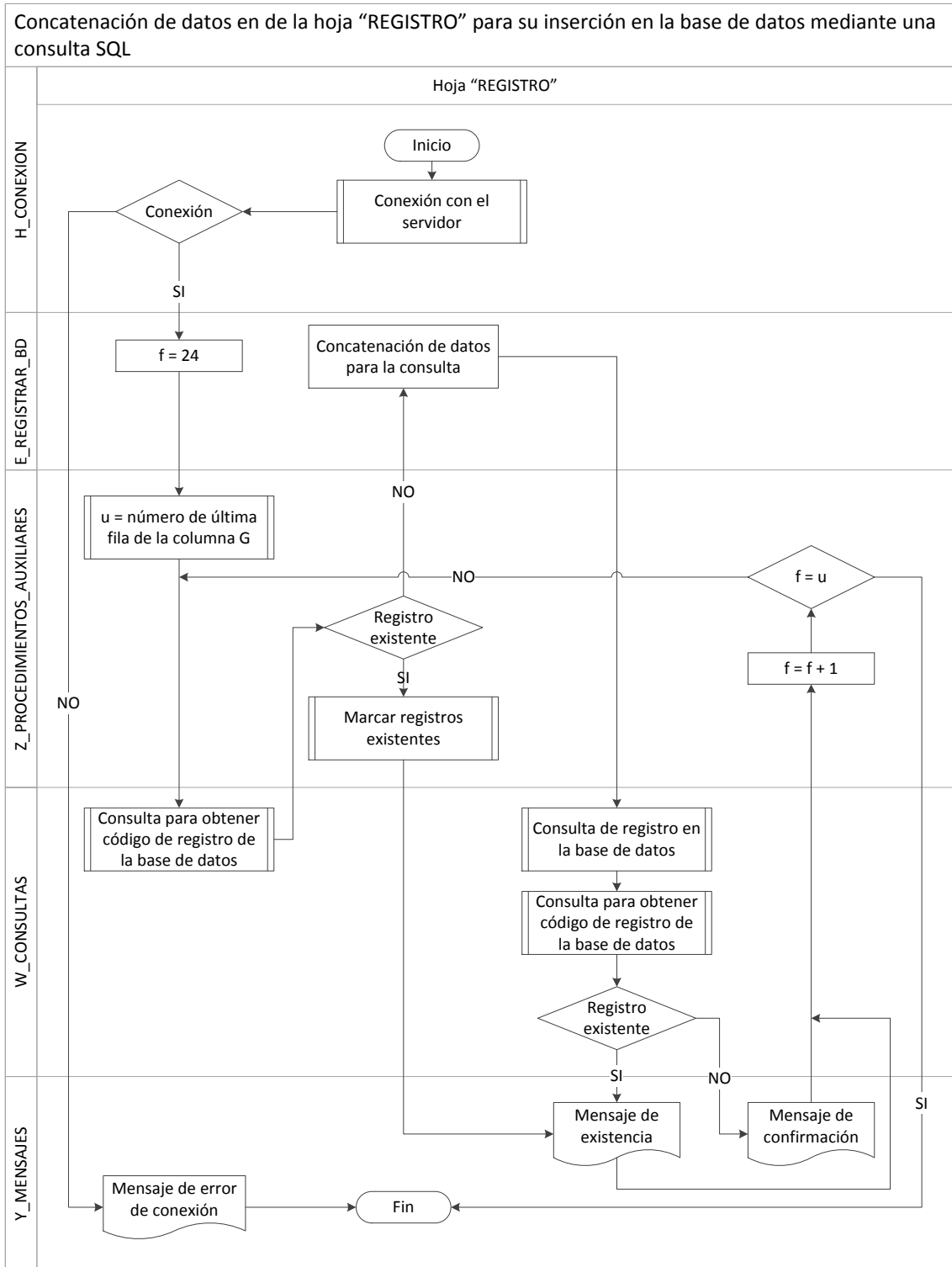
**Imagen 32: Diagrama de flujo del algoritmo de copia de un archivo**



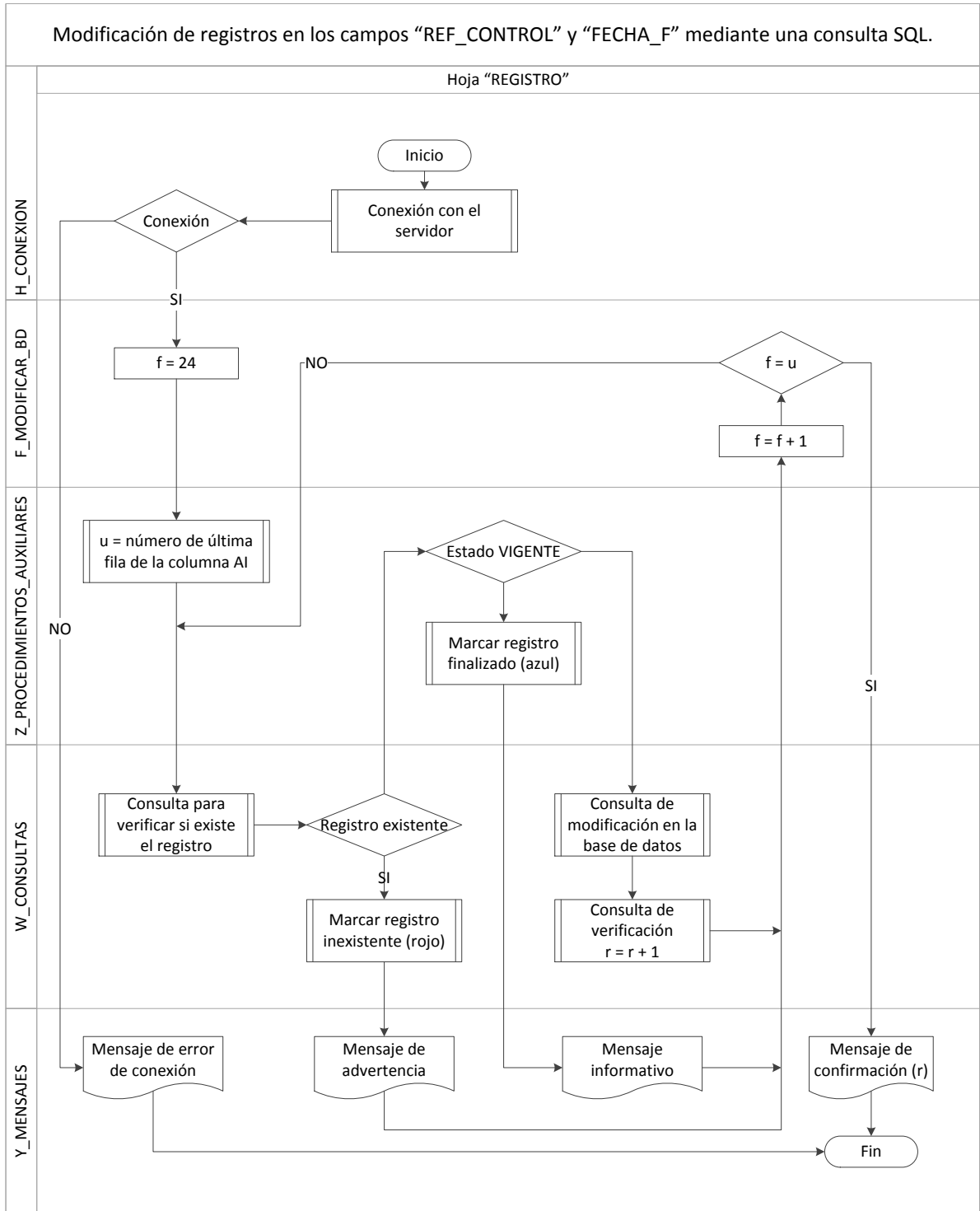
**Imagen 33: Diagrama de flujo del algoritmo de transferencia de datos**



**Imagen 34: Diagrama de flujo del algoritmo de consulta SQL (Registro)**

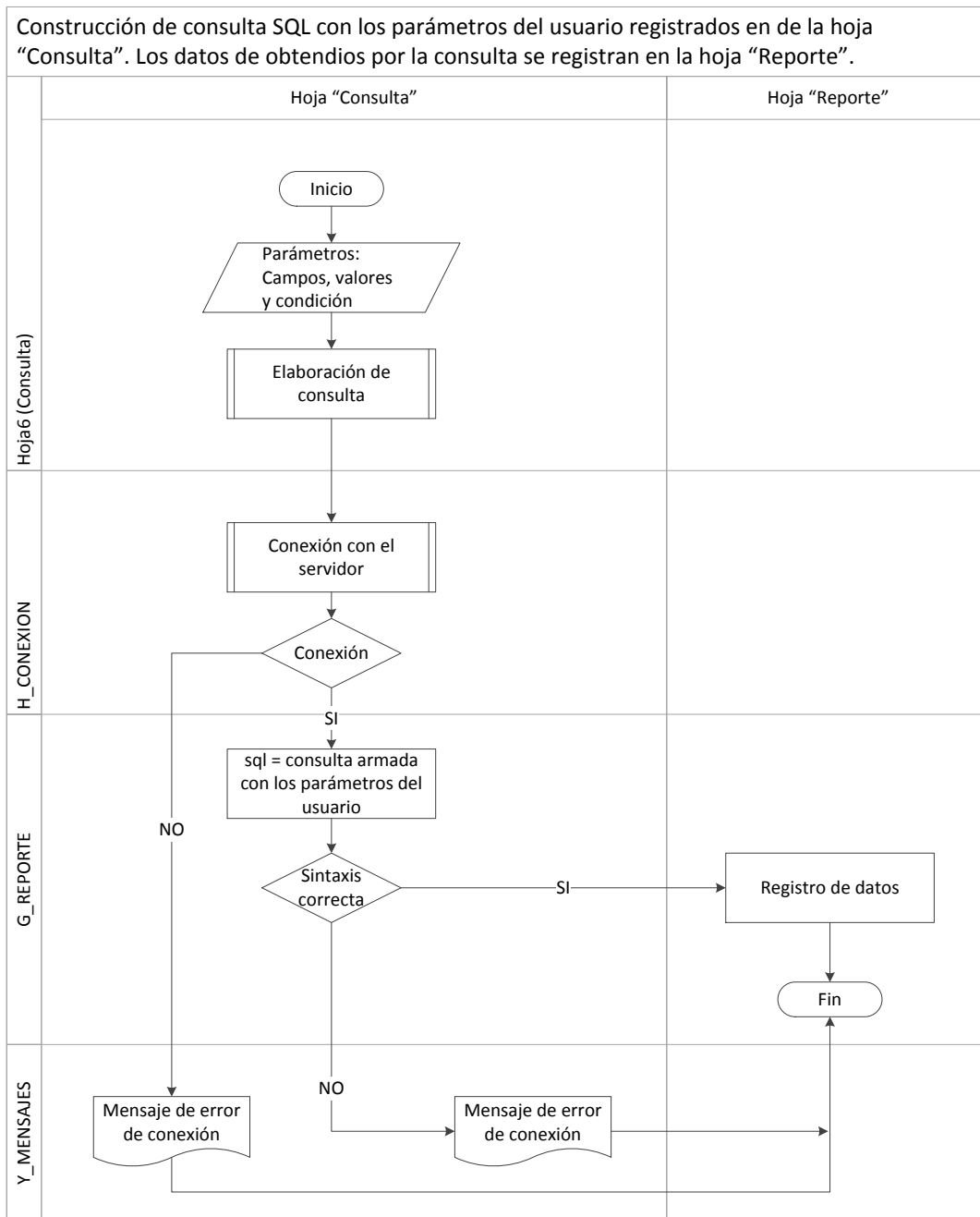


**Imagen 35: Diagrama de flujo del algoritmo de consulta SQL (Modificación)**





**Imagen 36: Diagrama de flujo del algoritmo de consulta SQL (Reporte)**



## Anexo 4: Descripción del código

**Tabla 43: Variables**

Variable	Tipo	Descripción
c	Integer	Número de columna.
f	Integer	Número de fila.
i	Integer	Índice de fila de la matriz MAT.
j	Integer	Índice de columna de la matriz MAT.
n	Integer	Número de una característica requerida.
r	Integer	Número de una característica requerida.
a	Integer	Número de una característica requerida.
b	Integer	Número de una característica requerida.
u	Integer	Número de última fila en un columna especificada.
fecha	Date	Fecha en formato dd/mm/yyyy.
t	String	Transacciones concatenadas.
v	String	Datos concatenados.
lista	String	Campos mostrados en el reporte.
campo	String	Campo utilizado en las cláusulas Where y Order By.
valor	String	Valores mostrados en el reporte.
codigo_exc	String	Valor en la columna AI (CODIGO) de la hoja "REGISTRO".
hh	String	Hora en la que se ejecuta el procedimiento.
mm	String	Minuto en el que se ejecuta el procedimiento.
hora	String	Código: hh_mm_ss
nc	String	Nombre de columna.
Nom_Arch	String	Nombre de archivo.
Nom_hoja	String	Nombre de la hoja.
Nom_Form	String	Nombre del FORMULARIO.
nombres	String	Nombres concatenados.
rango	String	Rango donde se aplica una instrucción.
color	String	Color de relleno de un rango.
titulo	String	Título del rango.
sql	String	Consulta SQL.
ubicacion	String	Carpeta donde está ubicado el FORMULARIO.
descripcion	String	Descripción de Microsoft sobre un error.
ARCHIVO	Variant	Valor del nombre de archivo
HOJA	Variant	Valor del nombre de hoja.
CELDA	Variant	Valor de celda.
DATOS	Variant	Valores de la hoja "RIESGO".
MAT	Variant	Matriz donde se almacenan los datos.
RIESGO	Variant	Valor de RIESGO.
USUARIO	Variant	Valor de USUARIO.

**Tabla 44: Variables públicas**

Variable	Tipo	Descripción
codigo_sql	String	Código de registro en la Base de Datos (Valor del campo CODIGO).
RefControl	String	Valor del campo REF_CONTROL.
CN	ADODB.Connection	Representación de la conexión con la Base de Datos.
RS	ADODB.Recordset	Representación de una tabla de registros.

**Tabla 45: Constantes**

Sintaxis
Public Const contraseña = ""
Public Const tabla = "REGISTRO_NEW"
Public Const h1 = "Configuración"
Public Const h2 = "COPIA"
Public Const h3 = "FORMULARIO"
Public Const h4 = "Aprobadores"
Public Const h5 = "REGISTRO"
Public Const h6 = "Consulta"
Public Const h7 = "Reporte"
Public Const o1 = "REGISTRAR_BD"
Public Const o2 = "MODIFICAR_BD"
Public Const o3 = "CREAR_COPIA"

**Tabla 46: Formato de colores para la descripción del código**

Color	Descripción
Texto	Referencia a instrucciones de procedimientos y funciones.
Texto	Referencia a operadores, objetos y sus propiedades .
Texto	Referencia a expresiones de cadena.
Texto	Referencia a valores numéricos.
Texto	Referencia a variables.
Texto	Referencia a funciones.
Texto	Referencia a procedimientos.

**Tabla 47: Código para la lista de archivos en la hoja "COPIA"**

Sintaxis	Descripción
<pre>Private Sub Worksheet_SelectionChange(ByVal Target As Range) On Error GoTo salir:      If Not Intersect(Target, Range("C3:G3")) Is Nothing Then         For Each ARCHIVO In Workbooks             nombres = nombres &amp; ", " &amp; ARCHIVO.Name         Next archivo         With Range("C3:G3").Validation             .Delete             .Add Type:=xlValidateList, _                 AlertStyle:=xlValidAlertStop, _                 Operator:=xlBetween, _                 Formula1:=nombres         End With     End If Exit sub salir:     Call mensaje_error(Err.Description) End Sub</pre>	<p>Si se selecciona el rango "C3:G3" se actualiza la lista de Archivos.</p> <p>Concatenación de los nombres de todos los archivos Excel abiertos en el escritorio.</p> <p>Rango donde se aplica la validación.</p> <p>Borrar valores previos de la lista.</p> <p>Validación de tipo lista con valores obtenidos en la instrucción <b>For Each</b>.</p> <p>Mensaje en caso de error.</p>

**Tabla 48: Código para la lista de hojas en la hoja “COPIA”**

Sintaxis	Descripción
<pre>Private Sub Worksheet_Change(ByVal Target As Range) On Error GoTo salir:      If Not Intersect(Target, Range("C3:G3")) Is Nothing Then         Nom_Arch = Range("C3").Value         For Each HOJA In Workbooks(Nom_Arch).Sheets             nombres = nombres &amp; ", " &amp; HOJA.Name         Next HOJA         With Range("C4:G4").Validation             .Delete             .Add Type:=xlValidateList, _                 AlertStyle:=xlValidAlertStop, _                 Operator:=xlBetween, _                 Formula1:=nombres         End With     End If Exit sub salir:     Call mensaje_error(Err.Description) End Sub</pre>	<p>Si se modifica el rango "C3:G3" se actualiza la lista de Hojas.</p> <p>Asignación del nombre de archivo.</p> <p>Concatenación de los nombres de todas las hojas del archivo Excel seleccionado en el rango "C3:G3" de la hoja "COPIA".</p> <p>Rango donde se aplica la validación.</p> <p>Borrar valores previos de la lista.</p> <p>Validación de tipo lista con valores obtenidos en la instrucción For Each.</p> <p>Mensaje en caso de error.</p>

**Tabla 49: Código para ejecutar el procedimiento “actualizar\_formulario”**

Sintaxis	Descripción
<pre>Private Sub Worksheet_Change(ByVal Target As Range) On Error GoTo salir:     If Not Intersect(Target, Range("A:N")) Is Nothing Then         Call actualizar_formulario     End If Exit Sub salir:     Call mensaje_error(Err.Description) End Sub</pre>	<p>Si se modifica el rango de columnas "A:N" se ejecuta el procedimiento actualizar_formulario.</p> <p>Mensaje en caso de error.</p>

**Tabla 50: Código para armar la consulta del reporte**

Sintaxis	Descripción
<pre>Private Sub Worksheet_Change(ByVal Target As Range) On Error GoTo salir:      If Not Intersect(Target, Range("B5:C50")) Is Nothing Then         u = ultima_fila("B")         lista = Range("B6").Value         If u &gt; 6 Then             For Each CELDA In Range("B7:B" &amp; u)                 lista = lista &amp; ", " &amp; CELDA.Value             Next CELDA         End If         u = ultima_fila("C")         valor = Range("C6").Value         If u &gt; 6 Then             For Each CELDA In Range("C7:C" &amp; u)                 valor = valor &amp; ", " &amp; CELDA.Value             Next CELDA         End If         campo = Range("C5").Value         Range("B3") = consulta_reporte(lista, campo, valor)     End If Exit sub salir:     Call mensaje_error(Err.Description) End Sub</pre>	<p>Si se modifica el rango "B5:C50" se registra la consulta del reporte.</p> <p>Asignación de la última fila de la columna "B".</p> <p>Asignación del primer campo.</p> <p>Si el número de la última fila es mayor a 6 (más de 1 campo) se concatenan todos los campos de la columna "B".</p> <p>Asignación de la última fila de la columna "C".</p> <p>Asignación del primer valor.</p> <p>Si el número de la última fila es mayor a 6 (más de 1 valor) se concatenan todos los valores de la columna "C".</p> <p>Asignación de condición.</p> <p>Registro de consulta.</p> <p>Mensaje en caso de error.</p>

**Tabla 51: Código para la copia de datos**

Sintaxis	Descripción
<pre> Sub A_COPIAR_DATOS() On Error GoTo salir:   Sheets(h2).Select   Range("A8:H507").Clear   Nom_Form = ActiveWorkbook.Name   Nom_Arch = Range("C3").Value   Nom_hoja = Range("C4").Value   f = Sheets(h1).Range("C3") + 1   For c = 1 To 8     nc = Sheets(h1).Cells(c + 5, 3)     On Error GoTo salir1:     Windows(Nom_Arch).Activate     On Error GoTo salir2:     Sheets(Nom_hoja).Select     On Error GoTo salir:     Range(nc &amp; f &amp; ":" &amp; nc &amp; 499 + f).Copy     Windows(Nom_Form).Activate     Call pegar(8, c)    Next c   Call revisar_formato Exit Sub salir:   Call mensaje_error(Err.Description) Exit Sub salir1:   Windows(Nom_Form).Activate   Call mensaje_error_archivo Exit Sub salir2:   Windows(Nom_Form).Activate   Call mensaje_error_hoja End Sub </pre>	<p>Selección de la hoja "COPIA".</p> <p>Borrar rango.</p> <p>Asignación del nombre de FORMULARIO.</p> <p>Asignación del nombre de Archivo.</p> <p>Asignación del nombre de Hoja.</p> <p>Asignación de fila inicial del rango a copiar.</p> <p>Asignación del nombre de columna.</p> <p>Activación de Archivo.</p> <p>Selección de Hoja.</p> <p>Copia de valores de la columna nc (500 celdas).</p> <p>Activación de FORMULARIO.</p> <p>Pegar en la fila 8, columna c.</p> <p>Revisión de formato para los datos copiados.</p> <p>Mensaje en caso de error.</p> <p>Activación de FORMULARIO.</p> <p>Mensaje en caso de error con el nombre de Archivo</p> <p>Activación de FORMULARIO.</p> <p>Mensaje en caso de error con el nombre de Hoja</p>

Tabla 52: Código para transferir datos al formato de la hoja "FORMULARIO"

Sintaxis	Descripción
<pre> Sub B_FORMATO_FORMULARIO() Dim MAT(20, 12) As Variant On Error GoTo salir:   Sheets(h3).Select   Call borrar_datos_formulario   Sheets(h2).Select   For f = 8 To ultima_fila("G")     n = contador(f)      If n = 6 Then i = i + 1: b = 0      Select Case n &amp; b       Case 60         For j = 1 To 6           MAT(i, j) = Cells(f, j + 1)         Next       Case 30         For j = 7 To 9           MAT(i, j) = Cells(f, j - 2)         Next         b = 1       Case 31         For j = 10 To 12           MAT(i, j) = Cells(f, j - 5)         Next       Case 10, 11         t = MAT(i, j - 1)         MAT(i, j - 1) = t &amp; ", " &amp; Cells(f, 7)     End Select   Next f   Sheets(h3).Select   For i = 1 To 20     For j = 1 To 12       Range("Riesgo_" &amp; i &amp; "_" &amp; j) = MAT(i, j)     Next j   Next i   Range("D16") = Sheets(h2).Range("A8")   Range("J7") = Sheets(h2).Range("H8")   Range("O11").Select Exit Sub salir:   Call mensaje_error(Err.Description) End Sub </pre>	<p>Matriz de 20 filas y 12 columnas para almacenar los datos.</p> <p>Selección de la hoja "FORMULARIO".</p> <p>Selección de la hoja "COPIA".</p> <p>Asignación del número de datos en la fila f</p> <p>Si n = 6 se incrementa el índice de fila de la matriz. Cuando b = 0 se almacena la función 1 y 2.</p> <p>Almacenamiento de: ID-Conflicto, Valoración, Descripción, Función 1, Descripción función 1, Transacción 1.</p> <p>Almacenamiento de: Función 2, Descripción función 2, Transacción 2.</p> <p>Cuando b = 1 se almacena la función 3.</p> <p>Almacenamiento de: Función 3, Descripción función 3, Transacción 3.</p> <p>Asignación del dato almacenado.</p> <p>Almacenamiento de transacciones concatenadas.</p> <p>Selección de la hoja "FORMULARIO".</p> <p>Registro en el rango Riesgo_i_j (Nombre definido para cada rango en la hoja "FORMULARIO").</p> <p>Registro del dato de Usuario.</p> <p>Registro del dato de Referencia 1.</p> <p>Selección celda "O11".</p> <p>Mensaje en caso de error.</p>

Tabla 53: Código para crear un copia del formulario

Sintaxis	Descripción
<pre> Sub C_CREAR_COPIA() On Error GoTo salir:     ubicacion = ActiveWorkbook.Path     Sheets(h3).Select     Sheets(h5).Range("A24:AI5023").Clear     Sheets(Array(h3, h4, h5)).Copy     Sheets(h5).Shapes.Range(Array(o1, o2)).Delete     Sheets(h3).Shapes.Range(Array(o3)).Delete     Sheets(Array(h4, h5)).Visible = False     Range("D16:J20").Interior.Pattern = xlNone     Range("O11").Select     Select Case Sheets(h4).Range("X8").Value         Case 1             Rows("73:382").EntireRow.Hidden = True         Case 2             Rows("135:382").EntireRow.Hidden = True         Case 3             Rows("197:382").EntireRow.Hidden = True         Case 4             Rows("259:382").EntireRow.Hidden = True         Case 5             Rows("321:382").EntireRow.Hidden = True     End Select     For n = 0 To 20         Call editar("Rango" &amp; n, "Editable_" &amp; n)     Next     ActiveSheet.Protect (contraseña), _         AllowFormattingCells:=True      Call guardar(ubicacion) Exit Sub salir:     Call mensaje_error(Err.Description) End Sub </pre>	<p>Asignación de la ubicación del FORMULARIO.          Selección de la hoja "FORMULARIO".          Borrar el rango en la hoja "REGISTRO".          Creación de nuevo archivo con las hojas "FORMULARIO", "Aprobadores" y "REGISTRO".          Eliminación de formas en la hoja "REGISTRO".          Eliminación de formas en la hoja "FORMULARIO".          Ocultar las hojas "Aprobadores" y "REGISTRO".          Fondo sin relleno en el rango de usuarios.          Selección de celda Área.</p> <p>Se ocultan las páginas según el número registrado en la hoja "Aprobadores" (celda X8).</p> <p>Inclusión de Rangos editables.</p> <p>Protección del nuevo archivo.</p> <p>Guardar el nuevo archivo en la ubicación del FORMULARIO</p> <p>Mensaje en caso de error.</p>



**Tabla 54: Código para transferir datos al formato de la hoja “REGISTRO”**

Sintaxis	Descripción
<pre> Sub D_FORMATO_REGISTRO() On Error GoTo salir:     Sheets(h2).Select     Nom_Form = ActiveWorkbook.Name     Nom_Arch = Range("C3").Value     Sheets(h5).Range("A24:AI5023").Clear     On Error GoTo salir1:     Windows(Nom_Arch).Activate     On Error GoTo salir:     Sheets(h5).Visible = True     Sheets(h5).Select     Range("A24:AI5023").Clear     For Each RIESGO In Range("E1:E21")         If RIESGO &lt;&gt; "" Then n = n + 1     Next RIESGO     For Each USUARIO In Range("AK2:AK21")         If USUARIO = "" Then Exit For         Range("B2") = USUARIO         Range("A2:AI" &amp; n).Copy         Call pegar(ultima_fila("A") + 1, 1)         Application.CutCopyMode = False     Next USUARIO     If Nom_Form &lt;&gt; Nom_Arch Then         u = ultima_fila("AI")         DATOS = Range("A24:AI" &amp; u)         Sheets(h5).Visible = False         Windows(Nom_Form).Activate         Sheets(h5).Select         Range("A24:AI" &amp; u) = DATOS     End If     Columns("X:X").NumberFormat = "dd/mm/yyyy;@"     Range("A24").Select Exit Sub salir:     Call mensaje_error(Err.Description) Exit Sub salir1:     Windows(Nom_Form).Activate     Call mensaje_error_archivo End Sub </pre>	<p>Selección de la hoja “COPIA”.</p> <p>Asignación del nombre de FORMULARIO.</p> <p>Asignación del nombre de Formulario Trabajado.</p> <p>Borrar datos en la hoja "REGISTRO" (Formulario).</p> <p>Activación de Formulario.</p> <p>Visualización de la hoja "REGISTRO".</p> <p>Selección de la hoja "REGISTRO".</p> <p>Borrar datos en la hoja "REGISTRO".</p> <p>Determinación del número de celdas no vacías en el rango "E1:E21".</p> <p>Si un valor de USUARIO está vacío se continua en la línea de código siguiente a “Next USUARIO”.</p> <p>Registro del valor de USUARIO.</p> <p>Copia de datos.</p> <p>Pegar en la fila siguiente al último dato de la columna "A"</p> <p>Termina el modo copia.</p> <p>Asignación de la última fila de la columna "AI".</p> <p>Asignación de datos.</p> <p>Ocultar hoja "REGISTRO" en el Formulario Trabajado.</p> <p>Activación de FORMULARIO.</p> <p>Selección de la hoja "REGISTRO".</p> <p>Registro en la hoja "REGISTRO".</p> <p>Formato para la columna X (dd/mm/yyyy) y selección de la celda A24.</p> <p>Mensaje en caso de error.</p> <p>Activación de FORMULARIO.</p> <p>Mensaje en caso de error con el nombre del Formulario Trabajado</p>

**Tabla 55: Código para insertar registros**

Sintaxis	Descripción
<pre> Sub E_REGISTRAR_BD() On Error GoTo salir:   Sheets(h5).Select   Range("A24:AI5023").Interior.Pattern = xlNone   Call Conexión   If CN.State = 1 Then     Set RS = New ADODB.Recordset     For f = 24 To ultima_fila("AI")       codigo_exc = Cells(f, 35).Value       Call obtener_datos_sql(codigo_exc)       If codigo_exc &lt;&gt; codigo_sql Then         v = Cells(f, 1)         For c = 2 To 35           If c &lt;&gt; 24 Then             v = v &amp; "','" &amp; Cells(f, c)           Else             fecha = Cells(f, 24)             v = v &amp; fecha_sql(fecha)             c = c + 1           End If         Next c         sql = consulta_insertar(v)         RS.Open sql, CN         Call obtener_datos_sql(codigo_exc)         If codigo_exc = codigo_sql Then           r = r + 1         End If       Else         Call marcar("AI" &amp; f, "amarillo")         Call mensaje_existente(codigo_sql)       End If     Next f     Call mensaje_insertados(f - 24, r)     Set RS = Nothing     CN.Close   Else     Call mensaje_error_conexion   End If Exit Sub salir:   Call mensaje_error(Err.Description) End Sub </pre>	<p>Selección de la hoja "REGISTRO". Eliminación del color de fondo.</p> <p>Si el estado de conexión está abierto se procede con el registro.</p> <p>Creación del objeto RS liberando referencias previas.</p> <p>Asignación de código en la fila f. Obtención de valores de CODIGO y REF_CONTROL Si los códigos son diferentes se procede con el registro. Asignación del primer dato.</p> <p>Concatenación de datos en la fila f (exceptuando la columna 24)</p> <p>En la columna 24 (FECHA_I) se concatena la función que convierte el formato de fecha Incremento del valor de c para omitir la columna 25 (FECHA_F).</p> <p>Asignación de la consulta para insertar los datos. Realización de la consulta. Obtención de valores de CODIGO y REF_CONTROL Si los códigos son iguales se contabiliza el registro.</p> <p>Marcar celda de color amarillo. Mensaje en caso de duplicidad.</p> <p>Mensaje del número de consultas y registros Liberación de recursos del sistema y memoria asociados con el objeto RS. Cierre de conexión.</p> <p>Mensaje en caso de error en la conexión.</p> <p>Mensaje en caso de error.</p>

**Tabla 56: Código para modificar registros**

Sintaxis	Descripción
<pre> Sub F_MODIFICAR_BD() On Error GoTo salir:   Sheets(h5).Select   Range("A24:AI5023").Interior.Pattern = xlNone   Call Conexión   If CN.State = 1 Then     Set RS = New ADODB.Recordset     For f = 24 To ultima_fila("AI")       codigo_exc = Cells(f, 35)       Call obtener_datos_sql(codigo_exc)       If codigo_exc = codigo_sql Then         If refControl = "VIGENTE" Then           sql = consulta_modificar(codigo_exc)           RS.Open sql, CN           Call obtener_datos_sql(codigo_exc)           If refControl = "FINALIZADO" Then             r = r + 1           End If         Else           Call marcar("AI" &amp; f, "azul")           Call mensaje_finalizado(codigo_exc)         End If       Else         Call marcar("AI" &amp; f, "rojo")         Call mensaje_inexistente(codigo_exc)       End If     Next f     Call mensaje_modificados(f - 24, r)     Set RS = Nothing     CN.Close   Else     Call mensaje_error_conexion   End If Exit Sub salir:   Call mensaje_error(Err.Description) End Sub </pre>	<p>Selección de la hoja "REGISTRO". Eliminación del color de fondo.</p> <p>Si el estado de conexión está abierto se procede con la modificación. Creación del objeto RS liberando referencias previas.</p> <p>Asignación del código de registro en la fila f. Obtención de valores de CODIGO y REF_CONTROL. Si los códigos son iguales se procede con la siguiente validación. Si el valor obtenido es "VIGENTE" se procede con la modificación. Asignación de la consulta para modificar el registro. Realización de la consulta. Obtención de valores de CODIGO y REF_CONTROL. Si el valor obtenido es "FINALIZADO" se contabiliza el registro. Marcar celda de color azul. Mensaje si el registro está finalizado.</p> <p>Marcar celda de color rojo. Mensaje si el registro no existe.</p> <p>Mensaje del número de consultas y registros. Liberación de recursos del sistema y memoria asociados con el objeto RS. Cierre de la conexión CN.</p> <p>Mensaje en caso de error en la conexión.</p> <p>Mensaje en caso de error</p>

**Tabla 57: Código para generar un reporte**

Sintaxis	Descripción
<pre> Sub G_REPORTE() On Error GoTo salir:   Sheets(h6).Select   Call Conexión    If CN.State = 1 Then      Set RS = New ADODB.Recordset     sql = Range("B3").Value     If UCase(Left(sql, 6)) &lt;&gt; "SELECT" _     Then GoTo salir1:     RS.Open sql, CN     Sheets(h7).Select     Range("A:AJ").Clear     For c = 1 To RS.Fields.Count       Cells(1, c) = RS.Fields(c - 1).Name     Next     f = 2     Do While Not RS.EOF       For c = 1 To RS.Fields.Count         Cells(f, c) = RS.Fields(c - 1)       Next       f = f + 1       RS.MoveNext     Loop     Columns("A:AJ").WrapText = False      Set RS = Nothing     CN.Close   Else     Call mensaje_error_conexion   End If Exit Sub salir1:   Call mensaje_advertencia Exit Sub salir:   Call mensaje_error(Err.Description) End Sub </pre>	<p>Selección de la hoja "Consulta".</p> <p>Si el estado de conexión está abierto se procede con la obtención del reporte.</p> <p>Creación del objeto RS liberando referencias previas.</p> <p>Asignación de consulta.</p> <p>Si la consulta no usa la sentencia <b>SELECT</b> se continua en la línea "salir1:".</p> <p>Realización de la consulta.</p> <p>Selección de la hoja "Reporte".</p> <p>Borrar rango.</p> <p>Registro de encabezados en la fila 1. (RS.Fields.Count = Número de campos del Recordset)</p> <p>Repetición mientras RS.EOF se falso.</p> <p>Registro de datos en la fila f.</p> <p>Incremento del valor de f.</p> <p>Cambio al siguiente registro del Recordset.</p> <p>Ajuste de texto.</p> <p>Liberación de recursos del sistema y memoria asociados con el objeto RS.</p> <p>Cierre de la conexión.</p> <p>Mensaje en caso de error en la conexión.</p> <p>Mensaje en caso de no utilizar la sentencia "SELECT"</p> <p>Mensaje en caso de error.</p>

**Tabla 58: Código de conexión al servidor**

Sintaxis	Descripción
<pre> Public Sub Conexión()   Set CN = New ADODB.Connection   On Error Resume Next    CN.ConnectionString = _     "Provider=SQLOLEDB.1;" &amp; _     "Integrated Security=SSPI;" &amp; _     "Data Source=Nombre Servidor;" &amp; _     "Database=Nombre Base de Datos"   CN.Open End Sub </pre>	<p>Creación del objeto CN liberando referencias previas.</p> <p>En caso de error se continua en la instrucción siguiente a donde se produjo el error.</p> <p>Cadena de conexión:</p> <p>Proveedor,</p> <p>Autenticación de Windows,</p> <p>Nombre del Servidor,</p> <p>Nombre de la Base de datos.</p> <p>Apertura de conexión.</p>

**Tabla 59: Código con la consulta para seleccionar datos**

Sintaxis VBA	Sintaxis SQL
<pre>Public Function consulta_seleccionar(codigo_exc As String)     consulta_seleccionar = _     "Select " &amp; _     "CODIGO" &amp; "," &amp; _     "REF_CONTROL" &amp; _     " From " &amp; _     tabla &amp; _     " Where " &amp; _     "CODIGO='" &amp; codigo_exc &amp; "'" &amp; _     ";" End Function</pre>	<pre>Select CODIGO, REF_CONTROL From REGISTRO_NEW Where CODIGO='valor_de_codigo' ;</pre>

**Tabla 60: Código con la consulta para insertar registros**

Sintaxis VBA	Sintaxis SQL
<pre>Public Function consulta_insertar(valor As String)     consulta_insertar = _     "Insert Into " &amp; _     tabla &amp; _     " Values ('" &amp; valor &amp; "'" &amp; _     ";" End Function</pre>	<pre>Insert Into REGISTRO_NEW Values('valor_1', ..., ['valor_n']) ;</pre>

**Tabla 61: Código con la consulta para modificar registros**

Sintaxis VBA	Sintaxis SQL
<pre>Public Function consulta_modificar(codigo_exc As String)     consulta_modificar = _     "Update " &amp; _     tabla &amp; _     " Set " &amp; _     "REF_CONTROL='FINALIZADO'," &amp; _     "FECHA_F=GETDATE()" &amp; _     " Where " &amp; _     "CODIGO='" &amp; codigo_exc &amp; "'" &amp; _     ";" End Function</pre>	<pre>Update REGISTRO_NEW Set REF_CONTROL='FINALIZADO', FECHA_F=GETDATE() Where CODIGO='valor_de_codigo' ;</pre>

**Tabla 62: Código con la consulta para generar un reporte**

Sintaxis VBA	Sintaxis SQL
<pre>Public Function consulta_reporte(lista As String, campo As String, valor As String)     consulta_reporte = _     "Select " &amp; _     lista &amp; _     " From " &amp; _     tabla &amp; _     " Where " &amp; _     campo &amp; " IN ('" &amp; valor &amp; "'" &amp; _     _     " Order By " &amp; _     campo &amp; _     ";" End Function</pre>	<pre>Select lista_de_campos From REGISTRO_NEW Where campo IN (lista_de_valores) Order By campo ;</pre>

**Tabla 63: Código para convertir el formato de fecha**

Sintaxis VBA	Descripción
<pre>Public Function fecha_sql(fecha As Date)     fecha_sql = "',CONVERT(DATE,'" &amp; fecha &amp; "','103),'"' End Function</pre>	<p>La función CONVERT() de SQL convierte el formato del parámetro <code>fecha</code> (dd/mm/yyyy) al formato yyyy-mm-dd.</p>

**Tabla 64: Código para obtener datos**

Sintaxis	Descripción
<pre>Public Sub obtener_datos_sql(codigo_exc As String)     sql = consulta_seleccionar(codigo_exc)     On Error GoTo salir:     RS.Open sql, CN     On Error Resume Next     codigo_sql = RS.Fields(0)     refControl = RS.Fields(1)     RS.Close Exit Sub salir:     Call mensaje_error(Err.Description) End Sub</pre>	<p>Asignación de la consulta para obtener datos</p> <p>Realización de la consulta.</p> <p>Asignación del valor de "CODIGO".</p> <p>Asignación del valor de "REF_CONTROL".</p> <p>Cierre de la conexión CN.</p> <p>Mensaje en caso de error.</p>

**Tabla 65: Código para actualizar la hoja "FORMULARIO"**

Sintaxis	Descripción
<pre>Sub actualizar_formulario()     u = ultima_fila("A")     ' Actualización de funciones:     Sheets(h3).Range("C11").formula = funcion(u, 2)     Sheets(h3).Range("H11").formula = funcion(u, 3)     Sheets(h3).Range("M11").formula = funcion(u, 4)     Sheets(h3).Range("D63").formula = funcion(u, 2)     Sheets(h3).Range("D64").formula = funcion(u, 3)     Sheets(h3).Range("H63").formula = funcion(u, 5)     Sheets(h3).Range("H64").formula = funcion(u, 6)     ' Actualización de listas desplegables:     Call lista_datos("O11", "A")     Call lista_datos("M8", "H")     For n = 1 To 20         Call lista_datos("Lista_" &amp; n &amp; "_1", "J")         Call lista_datos("Lista_" &amp; n &amp; "_2", "L")         Call lista_datos("Lista_" &amp; n &amp; "_3", "N")     Next End Sub</pre>	<p>Asignación de la última fila de la columna "A"</p> <p>Nombre Solicitante.</p> <p>Cargo Solicitante.</p> <p>Módulo Solicitante.</p> <p>Nombre Usuario Clave.</p> <p>Cargo Usuario Clave.</p> <p>Nombre Máxima Autoridad Nacional.</p> <p>Cargo Máxima Autoridad Nacional.</p> <p>Área.</p> <p>Empresa</p> <p>Tratamiento</p> <p>Tipo de evidencia</p> <p>Periodicidad</p>

**Tabla 66: Código para actualizar funciones**

Sintaxis
<pre>Public Function funcion(u As Integer, c As Integer)     'u = Última fila en la columna A (Área) de la hoja "Aprobadores".     'c = Número de columna en la matriz de búsqueda.     coincidir = "MATCH(O11," &amp; h4 &amp; "!A2:A" &amp; u &amp; ",0)"     indice = "INDEX(" &amp; h4 &amp; "!A2:F" &amp; u &amp; ",," &amp; coincidir &amp; "," &amp; c &amp; ")"     funcion = "=IFERROR(" &amp; indice &amp; ", """"")" End Function</pre>

**Tabla 67: Código para actualizar listas desplegables**

Sintaxis	Descripción
<pre>Public Sub lista_datos(rango As String, nc As String)     u = ultima_fila(nc)     With Sheets(h3).Range(rango).Validation         .Delete         .Add Type:=xlValidateList, _             AlertStyle:=xlValidAlertStop, _             Operator:=xlBetween, _             Formula1:="=" &amp; h4 &amp; "!" &amp; nc &amp; "2:" &amp; nc &amp; u     End With End Sub</pre>	<p>Asignación de la última fila de la columna especificada.</p> <p>Rango especificado donde se aplica la validación.</p> <p>Borrar valores previos de la lista.</p> <p>Validación de tipo lista con valores de la columna especificada.</p>

**Tabla 68: Código para mostrar mensaje de error (Nombre de archivo)**

Sintaxis	Descripción
<pre>Public Sub mensaje_error_archivo() MsgBox "Nombre de Archivo incorrecto.", _       vbOKOnly + vbCritical, _       "ERROR" End Sub</pre>	<p>Mensaje. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 69: Código para mostrar mensaje de error (Nombre de Hoja)**

Sintaxis	Descripción
<pre>Public Sub mensaje_error_hoja() MsgBox "Nombre de Hoja incorrecto.", _       vbOKOnly + vbCritical, _       "ERROR" End Sub</pre>	<p>Mensaje. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 70: Código para mostrar mensaje de error (Conexión)**

Sintaxis	Descripción
<pre>Public Sub mensaje_error_conexion() MsgBox "No se puede realizar la conexión.", _       vbOKOnly + vbCritical, _       "ERROR" End Sub</pre>	<p>Mensaje. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 71: Código para mostrar mensaje de error (Descripción)**

Sintaxis	Descripción
<pre>Public Sub mensaje_error(descripcion As String) MsgBox descripcion, _       vbOKOnly + vbCritical, _       "ERROR" End Sub</pre>	<p>Parámetro. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 72: Código para mostrar mensaje de advertencia (Sentencia Select)**

Sintaxis	Descripción
<pre>Public Sub mensaje_advertencia() MsgBox "Solo se admite la sentencia Select.", _       vbOKOnly + vbExclamation, _       "ADVERTENCIA" End Sub</pre>	<p>Mensaje. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 73: Código para mostrar mensaje informativo (Registros insertados)**

Sintaxis	Descripción
<pre>Public Sub mensaje_insertados(n As Integer, r As Integer) MsgBox "Consultas realizadas: " &amp; n &amp; _       vbNewLine &amp; vbNewLine &amp; _       "Registros insertados: " &amp; r, _       vbOKOnly + vbInformation, _       "INFORMACIÓN" End Sub</pre>	<p>Mensaje y Parámetro 1. Saltos de línea. Mensaje y Parámetro 2. Botón (Aceptar) e Icono. Titulo.</p>



**Tabla 74: Código para mostrar mensaje informativo (Registros modificados)**

Sintaxis	Descripción
<pre>Public Sub mensaje_modificados(n As Integer, r As Integer) MsgBox "Consultas realizadas: " &amp; n &amp; _       vbNewLine &amp; vbNewLine &amp; _       "Registros modificados: " &amp; r, _       vbOKOnly + vbInformation, _       "INFORMACIÓN" End Sub</pre>	<p>Mensaje y Parámetro 1. Saltos de línea. Mensaje y Parámetro 2. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 75: Código para mostrar mensaje informativo (Registro existente)**

Sintaxis	Descripción
<pre>Public Sub mensaje_existente(codigo_exc As String) MsgBox codigo_exc &amp; _       vbNewLine &amp; vbNewLine &amp; _       "Ya está registrado en la Base de Datos.", _       vbOKOnly + vbInformation, _       "INFORMACIÓN" End Sub</pre>	<p>Parámetro. Saltos de línea. Mensaje. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 76: Código para mostrar mensaje informativo (Registro inexistente)**

Sintaxis	Descripción
<pre>Public Sub mensaje_inexistente(codigo_exc As String) MsgBox codigo_exc &amp; _       vbNewLine &amp; vbNewLine &amp; _       "No está registrado en la Base de Datos.", _       vbOKOnly + vbInformation, _       "INFORMACIÓN" End Sub</pre>	<p>Parámetro. Saltos de línea. Mensaje. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 77: Código para mostrar mensaje informativo (Registro finalizado)**

Sintaxis	Descripción
<pre>Public Sub mensaje_finalizado(codigo_exc As String) MsgBox codigo_exc &amp; _       vbNewLine &amp; vbNewLine &amp; _       "Registro finalizado.", _       vbOKOnly + vbInformation, _       "INFORMACIÓN" End Sub</pre>	<p>Parámetro. Saltos de línea. Mensaje. Botón (Aceptar) e Icono. Titulo.</p>

**Tabla 78: Código para borrar datos**

Sintaxis	Descripción
<pre>Public Sub borrar_datos_formulario() Range("F7") = "REGISTRAR NÚMERO DE TICKET" For n = 0 To 10 Range("BORRAR_" &amp; n).Select Selection.ClearContents Next End Sub</pre>	<p>Registro de texto. Selección y borrado del rango BORRAR_i (Nombre definido para cada rango a borrar en la hoja "FORMULARIO").</p>

**Tabla 79: Código para revisar formato**

Sintaxis	Descripción
<pre> Sub revisar_formato()      u = ultima_fila("G")      For f = 8 To u         n = contador(f)         rango = "B" &amp; f &amp; ":G" &amp; f         Select Case n             Case 0, 2, 4, 5                 Call marcar(rango, "rojo")             Case 1                 If Cells(f, 7) = "" Then                     Call marcar(rango, "rojo")                 End If             Case 3                 If Cells(f, 5) = "" Or _                     Cells(f, 6) = "" Or _                     Cells(f, 7) = "" Then                     Call marcar(rango, "rojo")                 End If             Case 6                 a = f                 r = 0  bucle:                  a = a + 1                 v = Cells(a, 2)                 If v = "" And a &lt;= u Then GoTo bucle:                 For b = f To a - 1                     If Cells(b, 5) &lt;&gt; "" Then                         r = r + 1                     End If                 Next                 If r &lt; 2 Or r &gt; 3 Then                     rango = "E" &amp; f &amp; ":E" &amp; a - 1                     Call marcar(rango, "azul")                 End If             End Select             If Cells(f, 4).Value &lt;&gt; "" And _                 Right(Cells(f, 4).Value, 1) &lt;&gt; "." Then                 Call marcar("D" &amp; f, "amarillo")             End If         Next f         Range("A8").Select         Application.CutCopyMode = False     End Sub </pre>	<p>Asignación del número de última fila de la columna G</p> <p>Asignación del número de datos en la fila f</p> <p>Asignación de rango.</p> <p>El rango de la fila f se marca de color rojo</p> <p>Si la celda de la columna G está vacía el rango de la fila f se marca de color rojo</p> <p>Si una de las celdas de las columnas E, F y G está vacía el rango de la fila f se marca de color rojo.</p> <p>Asignación de valores.</p> <p>Contador del número de celdas vacías entre riesgos.</p> <p>Contador del número de funciones para un riesgo.</p> <p>Si el número de funciones es menor a 2 o mayor a 3 el rango de funciones de la columna E se marca de color azul.</p> <p>Si el texto de la columna D no termina en el signo ortográfico de punto (.) el rango de la fila f se marca de color amarillo.</p> <p>Terminar modo copia.</p>

**Tabla 80: Código para marcar el color de fondo de un rango**

Sintaxis	Descripción
<pre> Public Sub marcar(rango As String, color As String)     Select Case color         Case "rojo"             Range(rango).Interior.color = 255         Case "azul"             Range(rango).Interior.color = 16711680         Case "amarillo"             Range(rango).Interior.color = 65535     End Select End Sub </pre>	<p>Modificación del color de relleno según los parámetros de <b>rango</b> y <b>color</b> especificados.</p>

**Tabla 81: Código para guardar archivo**

Sintaxis	Descripción
<pre>Public Sub guardar(ubicacion As String)     hh = Hour(Now)     mm = Minute(Now)     ss = Second(Now)      If hh &lt; 10 Then hh = 0 &amp; hh     If mm &lt; 10 Then mm = 0 &amp; mm     If ss &lt; 10 Then ss = 0 &amp; ss     hora = hh &amp; "_" &amp; mm &amp; "_" &amp; ss      Application.DisplayAlerts = False      ActiveWorkbook.SaveAs Filename:= _     ubicacion &amp; "\" &amp; "Formulario " &amp; hora &amp; ".xlsx", _     FileFormat:=xlOpenXMLWorkbook, _     CreateBackup:=False      Application.DisplayAlerts = True End Sub</pre>	<p>Asignación de hora, minuto y segundo, obtenidos de la hora del sistema del equipo usado.</p> <p>Si el valor de una de las variables es menor a diez se concatena un cero a la izquierda.</p> <p>Concatenación de código (hh_mm_ss).</p> <p>Desactivación de la ventana de alerta.</p> <p>El archivo se guarda en la <b>ubicacion</b> especificada en formato "xlsx", con el nombre "Formulario hh_mm_ss".</p> <p>Activación de la ventana de alerta.</p>

**Tabla 82: Código para editar rangos**

Sintaxis	Descripción
<pre>Public Sub editar(titulo As String, rango As String)     ActiveSheet.Protection.AllowEditRanges.Add _     Title:=titulo, _     Range:=Range(rango) End Sub</pre>	<p>Inclusión de permisos para la edición según los parámetros de <b>titulo</b> (por ejemplo "Rango1") y <b>rango</b> (Nombre definido para cada rango editable en la hoja "FORMULARIO").</p>

**Tabla 83: Código que devuelve el número de datos en un rango**

Sintaxis	Descripción
<pre>Public Function contador(f As Integer)     contador = WorksheetFunction.CountA(Range("B" &amp; f &amp; ":G" &amp; f)) End Function</pre>	<p>Devuelve el número de celdas no vacías para cada <b>fila</b> en el rango "B:G".</p>

**Tabla 84: Código que devuelve el número de última fila**

Sintaxis	Descripción
<pre>Public Function ultima_fila(nc As String)     ultima_fila = Cells(Rows.Count, nc).End(xlUp).Row End Function</pre>	<p>Devuelve el número de la última celda no vacía en la <b>columna</b> especificada.</p>

**Tabla 85: Código para pegar datos**

Sintaxis	Descripción
<pre>Public Sub pegar(f As Integer, c As Integer)     Cells(f, c).PasteSpecial Paste:=xlPasteValues End Sub</pre>	<p>Procedimiento para pegar solo valores usando como referencia los parámetros <b>fila</b> y <b>columna</b> especificados.</p>

## Anexo 5: Atributos de evaluación de la calidad del software

**Tabla 86: Atributos de Funcionalidad**

Atributo	Criterio de evaluación del software	Cumplimiento	Justificación
Adaptabilidad	¿cuenta con un conjunto de funciones apropiadas para efectuar las tareas requeridas?	SI	Los módulos descritos en el Anexo 4, evidencian que se cumplen los requerimientos.
Exactitud	¿presenta resultados de acordes a las necesidades para las cuales fue creado?	SI	327 registros exitosos en el servidor de SOBOCE respaldan que los resultados son lo esperado.
Interoperabilidad	¿interactúa con otros sistemas previamente especificados?	SI	El formulario interactúa con la base de datos de SOBOCE mediante consultas SQL.
Conformidad	¿se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares?	N.A.	N.A.
Seguridad	¿previene el acceso no autorizado, ya sea accidental o premeditado?	SI	El formulario puede ser manipulado por cualquier usuario que tenga acceso. Pero el registro o modificación de datos está restringido por la autenticación del usuario (ver Tabla 58: Código de conexión al servidor)
<b>Cumplimiento = 4/4 = 100%</b>			

Fuente: Elaboración propia con base en la norma ISO 9126

**Tabla 87: Atributos de Confiabilidad**

Atributo	Criterio de evaluación del software	Cumplimiento	Justificación
Nivel de madurez	¿permite medir la frecuencia de fallas?	NO	El formulario permite identificar errores y gestionarlos, pero no cuenta con una funcionalidad específica para medir la frecuencia de fallas.
Tolerancia a fallas	¿mantiene un nivel específico de funcionamiento en caso de fallas?	SI	Al ser un archivo Excel en caso de un error o desconfiguración por causa del usuario es posible utilizar la versión original del formulario.
Recuperación	¿tiene la capacidad de restablecer el nivel de operación y recobrar los datos afectados por una falla?	SI	Si bien no es una funcionalidad desarrollada se aprovecha la característica de recuperación de las aplicaciones de Microsoft Office.
<b>Cumplimiento = 2/3 = 66%</b>			

Fuente: Elaboración propia con base en la norma ISO 9126

**Tabla 88: Atributos de Usabilidad**

Atributo	Criterio de evaluación del software	Cumplimiento	Justificación
Compresibilidad	¿los usuarios pueden reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación?	SI	El formulario cuenta con diagramas de flujo de los módulos principales (ver Anexo 3) y descripción detallada del código (ver Anexo 4)
Facilidad de aprendizaje	¿es fácil para los usuarios aprender a usar la aplicación?	SI	El formulario es intuitivo para los usuarios que conocen el proceso. Además cuenta con un manual y un video tutorial.
Operabilidad	¿es fácil para los usuarios la operación y el control?	SI	Cualquier usuario familiarizado con Excel puede interactuar con el formulario.
<b>Cumplimiento = 3/3 = 100%</b>			

Fuente: Elaboración propia con base en la norma ISO 9126

**Tabla A5 - 1: atributos de Eficiencia**

Atributo	Criterio de evaluación del software	Cumplimiento	Justificación
Comportamiento respecto al tiempo	¿son aceptables los tiempos de respuesta y de procesamiento de los datos?	SI	El procesamiento de datos y su registro en el formulario se realiza en segundos. El registro en la base de datos se realiza en un tiempo mayor debido a que la confirmación es un requerimiento solicitado.
Comportamiento respecto a los recursos	¿es aceptable la cantidad de recursos y la duración de su uso en la realización del procesamiento de datos?	SI	El recurso de memoria utilizado por el formulario es mínimo y su funcionamiento no presenta una carga excesiva para el procesador de la computadora.
<b>Cumplimiento = 2/2 = 100%</b>			

Fuente: Elaboración propia con base en la norma ISO 9126

**Tabla A5 - 2: Atributos de Mantenibilidad**

Atributo	Criterio de evaluación del software	Cumplimiento	Justificación
Análisis	¿es fácil diagnosticar las deficiencias o causas de falla?	SI	Las funcionalidades principales con los que interactúa el usuario cuentan con código para la gestión de errores facilitando su identificación.
Modificación	¿es fácil realizar modificaciones para que funcione en un ambiente diferente?	SI	El código del formulario se encuentra en módulos lo que facilita modificar un módulo si afectar a los demás.
Estabilidad	¿las modificaciones pueden realizarse sin el riesgo de efectos inesperados?	NO	Todo cambio representa un riesgo y no es posible asegurar que ocurran riesgos inesperados.
Facilidad de pruebas	¿son fáciles las pruebas de validación después de las modificaciones?	SI	Las pruebas son fáciles de realizar debido a que el editor de VBA permite realizar las pruebas paso a paso.
<b>Cumplimiento = 3/4 = 75%</b>			

Fuente: Elaboración propia con base en la norma ISO 9126

**Tabla A5 - 3: Atributos de Portabilidad**

<b>Atributo</b>	<b>Criterio de evaluación del software</b>	<b>Cumplimiento</b>	<b>Justificación</b>
Adaptabilidad	¿Es posible la adaptación a diferentes ambientes sin necesidad de aplicarle modificaciones?	NO	Por las características del lenguaje de programación el formulario solo funciona en la versión de escritorio de Microsoft Excel.
Facilidad de instalación	¿es fácil de instalar en un ambiente determinado?	N.A.	El formulario no requiere instalación.
Conformidad	¿se adhiere a estándares o convenciones relativas a portabilidad?	N.A.	El formulario no fue diseñado para ser portable.
Capacidad de reemplazo	¿puede ser fácilmente sustituido sin causar interrupciones significativas en el funcionamiento del sistema?	SI	El nuevo software debe tener una funcionalidad para conectarse al servidor de SOBOCE. Todas las funcionalidades pueden ser replicadas en cualquier con cualquier lenguaje de programación pero implica un costo mayor.
<b>Cumplimiento = 1/2 = 50 %</b>			

Fuente: Elaboración propia con base en la norma ISO 9126

## Anexo 6: Certificados de trabajo

<b>Empresa</b>	<b>Cargo</b>	<b>Año</b>
SOBOCE S.A.	Pasante	2017
SOBOCE S.A.	Analista de Procesos Junior	2018
SOBOCE S.A.	Analista del Centro de Información de Fábrica	2018
SOBOCE S.A.	Analista de Procesos Junior	2019





Sociedad Boliviana de Cemento S.A.

## CERTIFICADO DE PASANTIA

SGNRH-116/17

### A quien corresponda:

El suscrito Jefe Nacional Administrativo de Recursos Humanos de la empresa Sociedad Boliviana de Cemento S.A.

### CERTIFICA:

Que el señor **Juan Vidal Cordero Tintaya**, con cédula de identidad 7041288 LP, realizó una pasantía en nuestra empresa en el área de Tecnología y Procesos, desde el 11 de septiembre hasta el 31 de octubre de 2017, apoyando en el relevamiento de procesos diseñados y adoptados por la implementación del sistema SAP HANA.

Durante su presencia en la empresa, el señor Cordero demostró interés, compromiso y capacidad sobresalientes.

Es cuanto se certifica en estricta sujeción a la verdad.

La Paz, 13 de diciembre de 2017

Carlos Sandoval Navia  
**JEFE NACIONAL ADMINISTRATIVO  
DE RECURSOS HUMANOS**



C.C.: Archivo

Contacto Soboce  
**800-103-606**  
Línea gratuita

[www.soboce.com](http://www.soboce.com)

Planta Viacha: Telfs.: (591-2) 280-0101 / 280-0116 • Fax: (591-2) 280-0284  
Planta Ready Mix: Telf.: (591-2) 221-1272 • Fax: (591-2) 221-8703  
Comercialización: Calle Mercado 1075 • Telf.: (591-2) 240-6465 • Fax: (591-2) 240-7557  
Oficina Central: Calle Mercado 1075 • Telf. Piloto: (591-2) 240-6040 • Fax: (591-2) 240-7440 • Casilla 557  
La Paz - Bolivia







Sociedad Boliviana de Cemento S.A.

## CERTIFICADO DE TRABAJO

SGNRH-093/18

### **A quien corresponda:**

El suscrito Jefe Nacional Administrativo de Recursos Humanos de la empresa Sociedad Boliviana de Cemento S.A.

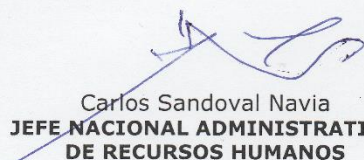
### **CERTIFICA:**

Que el señor **Juan Vidal Cordero Tintaya**, con cédula de identidad 7041288 LP, trabajó en la empresa bajo la modalidad de contrato a plazo fijo en los siguientes periodos:

- Desde el 18/01/2018 hasta el 20/04/2018, desempeñando el cargo de Analista de Procesos Junior
- Desde el 02/05/2018 hasta el 05/10/2018, desempeñando el cargo de Analista del CIF

Certificación que se extiende a petición del interesado y en estricta sujeción a la verdad.

La Paz, 3 de enero de 2019

  
Carlos Sandoval Navia  
**JEFE NACIONAL ADMINISTRATIVO  
DE RECURSOS HUMANOS**



c.c.: File personal pasivo

Contacto Soboce  
**800-103-606**  
Línea gratuita  
[www.soboce.com](http://www.soboce.com)

Planta Viacha: Telfs.: (591-2) 280-0101 / 280-0116 • Fax: (591-2) 280-0284  
Planta Ready Mix: Telf.: (591-2) 221-1272 • Fax: (591-2) 221-8703  
Comercialización: Calle Mercado 1075 • Telf.: (591-2) 240-6465 • Fax: (591-2) 240-7557  
Oficina Central: Calle Mercado 1075 • Telf. Piloto: (591-2) 240-6040 • Fax: (591-2) 240-7440 • Casilla 557  
La Paz - Bolivia



Cemento Viacha y Ready Mix son marcas de la Sociedad Boliviana de Cemento, SOBOCE S.A.





Sociedad Boliviana de Cemento S.A.

## CERTIFICADO DE TRABAJO

**SGNRH-059/19**

### **A quien corresponda:**

El suscrito Jefe Nacional Administrativo de Recursos Humanos de la empresa Sociedad Boliviana de Cemento S.A.

### **CERTIFICA:**

Que el señor **Juan Vidal Cordero Tintaya**, con cédula de identidad 7041288 LP, trabajó en la empresa bajo la modalidad de contrato a plazo fijo, desde el 04 de diciembre de 2018 hasta el 29 de marzo de 2019, habiendo desempeñado el cargo de Analista de Procesos Junior.

Certificación que se extiende a petición del interesado y en estricta sujeción a la verdad.

La Paz, 16 de septiembre de 2019

Carlos Sandoval Navia  
**JEFE NACIONAL ADMINISTRATIVO  
DE RECURSOS HUMANOS**



Contacto Soboce  
**800-103-606**  
Línea gratuita  
[www.soboce.com](http://www.soboce.com)

Planta Viacha: Telfs.: (591-2) 280-0101 / 280-0116 • Fax: (591-2) 280-0284  
Planta Ready Mix: Telf.: (591-2) 221-1272 • Fax: (591-2) 221-8703  
Comercialización: Calle Mercado 1075 • Telf.: (591-2) 240-8465 • Fax: (591-2) 240-7557  
Oficina Central: Calle Mercado 1075 • Telf. Piloto: (591-2) 240-6040 • Fax: (591-2) 240-7440 • Casilla 557  
La Paz - Bolivia



Cemento Viacha y Ready Mix son marcas de la Sociedad Boliviana de Cemento, SOBOCE S.A.





ESTADO PLURINACIONAL DE BOLIVIA

MINISTERIO DE DESARROLLO PRODUCTIVO Y ECONOMÍA PLURAL



2 0 2 3 - T T E S - 1 1 9 4 - 0 - 1

DIRECCIÓN DE DERECHO DE AUTOR Y DERECHOS CONEXOS  
RESOLUCIÓN ADMINISTRATIVA NRO. 1-2867/2023  
La Paz, 9 de Octubre del 2023

VISTOS:

La solicitud de Inscripción de Derecho de Autor presentada en fecha 2 de Octubre del 2023, por JUAN VIDAL CORDERO TINTAYA con C.I. N° 7041288 LP, con número de trámite DA-E 300327/2023, señala la pretensión de inscripción de la Memoria Laboral: "AUTOMATIZACIÓN DEL FORMULARIO PARA EL PROCEDIMIENTO DE GESTIÓN DE RIESGOS PARA EL ALTA O MODIFICACIÓN DE ROLES DE USUARIOS DEL SISTEMA SAP EN SOBOCE S.A.", cuyos datos y antecedentes se encuentran adjuntos y expresados en el Formulario de Declaración Jurada.

CONSIDERANDO

Que, en observación al Artículo 4° del Decreto Supremo N° 27938 modificado parcialmente por el Decreto Supremo N° 28152 el "Servicio Nacional de Propiedad Intelectual SENAPI, administra en forma desconcentrada e integral el régimen de la Propiedad Intelectual en todos sus componentes, mediante una estricta observancia de los regímenes legales de la Propiedad Intelectual, de la vigilancia de su cumplimiento y de una efectiva protección de los derechos de exclusiva referidos a la propiedad industrial, al derecho de autor y derechos conexos; constituyéndose en la oficina nacional competente respecto de los tratados internacionales y acuerdos regionales suscritos y adheridos por el país, así como de las normas y regímenes comunes que en materia de Propiedad Intelectual se han adoptado en el marco del proceso andino de integración".

Que, el Artículo 16° del Decreto Supremo N° 27938 establece "Como núcleo técnico y operativo del SENAPI funcionan las Direcciones Técnicas que son las encargadas de la evaluación y procesamiento de las solicitudes de derechos de propiedad intelectual, de conformidad a los distintos regímenes legales aplicables a cada área de gestión". En ese marco, la Dirección de Derecho de Autor y Derechos Conexos otorga registros con carácter declarativo sobre las obras del ingenio cualquiera que sea el género o forma de expresión, sin importar el mérito literario o artístico a través de la inscripción y la difusión, en cumplimiento a la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina, Ley de Derecho de Autor N° 1322, Decreto Reglamentario N° 23907 y demás normativa vigente sobre la materia.

Que, la solicitud presentada cumple con: el Artículo 6° de la Ley N° 1322 de Derecho de Autor, el Artículo 26° inciso a) del Decreto Supremo N° 23907 Reglamento de la Ley de Derecho de Autor, y con el Artículo 4° de la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina.

Que, de conformidad al Artículo 18° de la Ley N° 1322 de Derecho de Autor en concordancia con el Artículo 18° de la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina, referentes a la duración de los Derechos Patrimoniales, los mismos establecen que: "la duración de la protección concedida por la presente ley será para toda la vida del autor y por 50 años después de su muerte, a favor de sus herederos, legatarios y cesionarios".



"2023 AÑO DE LA JUVENTUD HACIA EL BICENTENARIO"

Oficina Central - La Paz  
Av. Montes, N° 575,  
entre Esq. Uruguay y  
C. Batallón Illimani.  
Telfs.: 2195700  
2194276 - 2195251

Oficina - Santa Cruz  
Av. Uruguay, Calle  
prolongación Quijarro,  
N° 29, Edif. Bicentenario.  
Telfs.: 3121752 - 72042936

Oficina - Cochabamba  
Calle Bolívar, N° 737,  
entre 16 de Julio y Antezana.  
Telfs.: 4411409 - 72042957

Oficina - El Alto  
Av. Juan Pablo II, N° 2560  
Edif. Multicentro El Ceibo  
Lda. Piso 2, Of. 58,  
Zona 16 de Julio.  
Telfs.: 2141001 - 72043029

Oficina - Chuquisaca  
Calle Kilómetro 7, N° 366  
casi esq. Urrutiolagoitia,  
Zona Parque Bolívar.  
Telf.: 72005873

Oficina - Tarija  
Av. La Paz, entre  
Calles Ciro Trigo y Avaroa  
Edif. Santa Clara, N° 243.  
Telf.: 72005286

Oficina - Oruro  
Calle 6 de Octubre N° 5837  
entre Ayacucho y Junín,  
Galería Central, Of. 14.  
Telf.: 67201288

Oficina - Potosí  
Av. Villazón entre calles  
Wenceslao Alba y San Alberto,  
Edif. AM. Salinas N° 212,  
Primer Piso, Of. 17.  
Telf.: 72018160

www.senapi.gob.bo





ESTADO PLURINACIONAL DE BOLIVIA

MINISTERIO DE DESARROLLO PRODUCTIVO Y ECONOMÍA PLURAL

Que, se deja establecido en conformidad al Artículo 4º de la Ley Nº 1322 de Derecho de Autor, y Artículo 7º de la Decisión 351 Régimen Común sobre Derecho de Autor y Derechos Conexos de la Comunidad Andina que: "...No son objeto de protección las ideas contenidas en las obras literarias, artísticas, o el contenido ideológico o técnico de las obras científicas ni su aprovechamiento industrial o comercial".

Que, el artículo 4, inciso e) de la ley 2341 de Procedimiento Administrativo, instituye que: "... en la relación de los particulares con la Administración Pública, se presume el principio de buena fe. La confianza, la cooperación y la lealtad en la actuación de los servidores públicos y de los ciudadanos ...", por lo que se presume la buena fe de los administrados respecto a las solicitudes de registro y la declaración jurada respecto a la originalidad de la obra.

**POR TANTO**

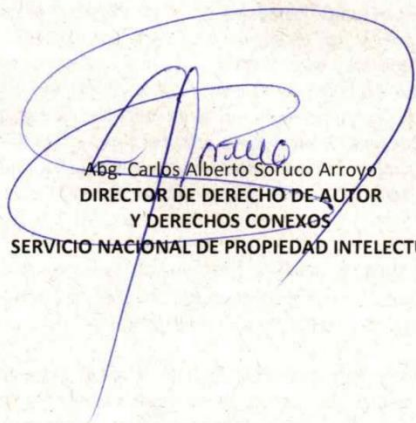
El Director de Derecho de Autor y Derechos Conexos sin ingresar en mayores consideraciones de orden legal, en ejercicio de las atribuciones conferidas



**RESUELVE:**

**INSCRIBIR** en el Registro de Tesis, Proyectos de Grado, Monografías y Otras Similares de la Dirección de Derecho de Autor y Derechos Conexos, la Memoria Laboral titulada: "**AUTOMATIZACIÓN DEL FORMULARIO PARA EL PROCEDIMIENTO DE GESTIÓN DE RIESGOS PARA EL ALTA O MODIFICACIÓN DE ROLES DE USUARIOS DEL SISTEMA SAP EN SOBOCE S.A.**", a favor del autor y titular: **JUAN VIDAL CORDERO TINTAYA** con C.I. **Nº 7041288 LP**, quedando amparado su derecho conforme a Ley, salvando el mejor derecho que terceras personas pudiesen demostrar.

Regístrese, Comuníquese y Archívese.

  
Abg. Carlos Alberto Soruco Arroyo  
**DIRECTOR DE DERECHO DE AUTOR  
Y DERECHOS CONEXOS**  
**SERVICIO NACIONAL DE PROPIEDAD INTELECTUAL**

CASA/ata  
c.c Arch.



**"2023 AÑO DE LA JUVENTUD HACIA EL BICENTENARIO"**

Oficina Central - La Paz  
Av. Montes, Nº 515,  
entre Esq. Uruguay y  
C. Batallón Illimani.  
Telfs.: 2195700  
2195276 - 2195251

Oficina - Santa Cruz  
Av. Uruguay, Calle  
prolongación Quijano,  
Nº 29, Edif. Bicentenario.  
Telfs.: 312752 - 72042936

Oficina - Cochabamba  
Calle Bolívar, Nº 737,  
entre 16 de Julio y Antezana.  
Telfs.: 4441493 - 72042957

Oficina - El Alto  
Av. Juan Pablo II, Nº 2560  
Edif. Multicentro El Ceibo  
Uda. Piso 2, Of. 58,  
Zona 16 de Julio.  
Telfs.: 2141001 - 72043029

Oficina - Chuquisaca  
Calle Kilómetro 7, Nº 366  
casi esq. Urrutogoitia,  
Zona Parque Bolívar.  
Telf.: 72005873

Oficina - Tarija  
Av. La Paz, entre  
Calles Ciro Trigo y Avanca  
Edif. Santa Clara, Nº 243.  
Telf.: 72095386

Oficina - Oruro  
Calle 6 de Octubre Nº 5837  
entre Ayacucho y Junín,  
Galería Central, Of. 14.  
Telf.: 6210188

Oficina - Potosí  
Av. Villazón entre calles  
Wenceslao Alba y San Alberto,  
Edif. AM. Salinas Nº 262,  
Primer Piso, Of. 11.  
Telf.: 72018160

[www.senapi.gov.bo](http://www.senapi.gov.bo)

**Autor:** Juan Vidal Cordero Tintaya

**Correo:** [juanvidaling@gmail.com](mailto:juanvidaling@gmail.com)

**Celular:** 65687332