

**UNIVERSIDAD CENTRAL
VICERRECTORÍA ACADÉMICA**

ESCUELA DE INFORMÁTICA Y ARQUITECTURA

**PROYECTO DE DESARROLLO DEL GESTOR DE ACTIVOS
DIGITALES PARA EL ÁREA DE SOPORTE TÉCNICO EN
INSTACREDIT, PODRÍAMOS LLAMARLO "ASSETPRO:
SOLUCIÓN INTEGRAL PARA LA GESTIÓN DE ACTIVOS
DIGITALES EN SOPORTE TÉCNICO"**

**MODALIDAD DE TESIS PARA OPTAR POR EL GRADO DE BACHILLERATO EN INGENIERÍA
INFORMÁTICA**

ESTUDIANTE: BRYAN JOSUÉ ARGÜELLO SABORÍO

TUTOR: ING. MAURICIO TORRES

SEDE METROPOLITANA, COSTA RICA

Marzo, 2025

Tabla de contenido

DEDICATORIA Y AGRADECIMIENTO	XI
RESUMEN	XII
ABSTRACT	XIII
CAPÍTULO I. PROBLEMA	1
PLANTEAMIENTO DEL PROBLEMA	2
OBJETIVO.....	3
OBJETIVO GENERAL.....	3
<i>Objetivos específicos</i>	3
JUSTIFICACIÓN	4
ANTECEDENTES	5
<i>Nacionales</i>	5
<i>Internacionales</i>	7
PROYECCIONES.....	9
<i>Alcances</i>	9
<i>Limitaciones</i>	10
CAPÍTULO II. MARCO TEÓRICO	12
HISTORIA DE LA PROGRAMACIÓN	13
<i>Definición</i>	13
<i>Antecedentes</i>	14
<i>Para las décadas de 1960 y 1970, se definieron varios paradigmas:</i>	16
<i>Década de 1980:</i>	18
<i>Década de 1990:</i>	18
<i>Desde el año 2000:</i>	18
FUNDAMENTOS DE LA PROGRAMACIÓN	19
<i>¿Qué es un lenguaje de programación?</i>	20
<i>Tipos de lenguajes de programación</i>	21
<i>Paradigmas de programación</i>	24
<i>Algoritmos de programación</i>	27
<i>Sistema de gestión de activos digitales</i>	28
<i>Métodos para el control de activos digitales</i>	30
SISTEMAS DE INFORMACIÓN APLICADOS A LA GESTIÓN DE ACTIVOS DIGITALES	37
<i>Sistemas de procesamiento de transacciones (TPS)</i>	38
<i>Sistemas de información gerencial (MIS)</i>	39

<i>Sistemas de control de procesos de negocio (BPM)</i>	40
<i>Sistemas de información de marketing (SIM)</i>	41
<i>Sistemas de colaboración empresarial (ERP)</i>	42
<i>Sistemas de soporte a la toma de decisiones (DSS)</i>	43
<i>Planificación</i>	45
<i>Análisis de requerimientos</i>	46
<i>Diseño</i>	46
<i>Desarrollo</i>	46
<i>Prueba</i>	46
<i>Integración y ejecución</i>	46
<i>Operación y mantenimiento</i>	47
REQUERIMIENTOS DE SISTEMA	48
<i>Requerimientos funcionales</i>	48
<i>Requerimientos no funcionales</i>	49
<i>Arquitectura</i>	52
<i>Estructura de datos</i>	54
<i>Interfaz</i>	57
<i>Componentes</i>	58
<i>Diagrama UML</i>	58
CASOS DE USO	60
<i>Generar reporte de auditoría</i>	61
<i>Crear expediente de soporte</i>	61
BASE DE DATOS EN UN GESTOR DE ACTIVOS DIGITALES	63
<i>¿Qué es Microsoft SQL Server?</i>	63
METODOLOGÍA DE DESARROLLO	64
<i>Metodologías recomendadas</i>	64
<i>Incremental</i>	65
CASOS DE USO CLAVE	65
BASE DE DATOS	66
<i>Historia de las bases de datos: un viaje a través del tiempo</i>	66
<i>Los precursores: sistemas de ficheros</i>	67
METODOLOGÍAS DE DESARROLLO	72
<i>Metodologías de desarrollo de software tradicionales</i>	73
<i>Metodologías ágiles</i>	74
¿QUÉ ES WINDOWS FORM?	76
¿QUÉ ES UN FORMULARIO?	76
CUATRO PASOS PARA CREAR UNA APLICACIÓN BÁSICA DE WINDOWS FORMS	78

<i>Paso 1. Diseñar la arquitectura de la aplicación</i>	78
<i>Paso 2. Implementar la lógica del modelo</i>	81
HERRAMIENTAS INGENIERILES	85
<i>Diagrama de Ishikawa</i>	85
IDENTIFICACIÓN DE LA EMPRESA.....	87
<i>Círculo virtuoso de cooperación</i>	87
ANTECEDENTES HISTÓRICOS.....	88
UBICACIÓN GEOGRÁFICA.....	89
ESTRUCTURA ORGANIZACIONAL	90
CANTIDAD DE EMPLEADOS	92
TIPOS DE PRODUCTOS	92
MERCADO DE EXPORTACIÓN	94
CAPÍTULO III. MARCO METODOLÓGICO	95
○ <i>Diseño de investigación</i>	96
○ <i>Métodos de recopilación de datos</i>	96
○ <i>Población y muestra</i>	96
○ <i>Procedimiento</i>	96
○ <i>Instrumentos de medición</i>	96
○ <i>Análisis de datos</i>	96
○ <i>Ética:</i>	97
IMPORTANCIA DEL MARCO METODOLÓGICO	97
○ <i>Organización y estructura:</i>	97
○ <i>Rigurosidad científica</i>	98
○ <i>Reproducibilidad y generalización</i>	98
○ <i>Claridad en la planificación</i>	98
○ <i>Énfasis en la pertinencia</i>	98
○ <i>Control de variables</i>	98
○ <i>Ética y responsabilidad</i>	99
TIPOS DE ENFOQUES	103
<i>¿Cuáles son los enfoques principales de investigación?</i>	103
<i>Tipos de investigación mixta</i>	107
DISEÑO DE LA INVESTIGACIÓN	109
<i>¿Para qué realizar un diseño de investigación?</i>	110
FUENTES DE INFORMACIÓN	111
• <i>Fuentes primarias</i>	111
• <i>Fuentes secundarias:</i>	112

• <i>Fuentes terciarias</i>	112
<i>Fuentes de información confiables</i>	112
¿QUÉ ES UNA VARIABLE?	114
REQUISITOS FUNCIONALES.....	115
REQUERIMIENTOS NO FUNCIONALES	116
CASOS DE USO.....	117
○ <i>Diagramas de flujo</i>	117
○ <i>Casos de uso</i>	117
CAPÍTULO IV. ANÁLISIS DE RESULTADOS.....	121
PLAN DE GESTIÓN DE RIESGOS	122
<i>Componentes principales</i>	123
<i>Matriz de riesgos</i>	125
FUENTE: (ARGÜELLO SABORÍO, B. (2025). RESULTADOS DE MAPA DE CALOR).	128
ESTUDIO DE FACTIBILIDAD.....	129
<i>Factibilidad técnica</i>	130
<i>Factibilidad económica</i>	131
<i>Factibilidad operativa</i>	133
<i>Descripción del proyecto</i>	135
<i>Objetivos del proyecto</i>	136
<i>Recursos necesarios</i>	136
<i>Evaluación de riesgo</i>	136
<i>Evaluación de la viabilidad operativa</i>	136
METODOLOGÍA UTILIZADA.....	138
• <i>Visualizar el trabajo</i>	138
• <i>Limitar el trabajo en curso</i>	138
• <i>Gestionar el flujo de trabajo</i>	138
• <i>Implementar políticas de procesos explícitas</i>	138
• <i>Implementar ciclos de comentarios</i>	139
• <i>Mejorar colaborando y evolucionar experimentando</i>	139
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	141
CONCLUSIONES.....	142
RECOMENDACIONES.....	143
CAPÍTULO VI. ANÁLISIS DE REQUERIMIENTOS.....	144
ANÁLISIS DE REQUERIMIENTOS.....	145
REQUERIMIENTOS DE USUARIO.....	145

REQUERIMIENTOS FUNCIONALES	146
REQUERIMIENTOS NO FUNCIONALES	146
REQUERIMIENTOS FUNCIONALES	153
REQUERIMIENTOS NO FUNCIONALES	158
MODELOS UML	161
<i>Casos de uso</i>	161
<i>Caso de uso 1. Login</i>	162
2. <i>Caso de uso: registro de usuarios</i>	163
3. <i>Caso de uso: registro de activos</i>	164
<i>Caso de uso 4: registro de proveedores</i>	165
<i>Caso de uso 5: registro de boletas de movimiento</i>	166
<i>Caso de uso 6: registro de expedientes</i>	168
<i>Caso de uso 7: Auditoría</i>	169
<i>Caso de uso 8: bitácora inicio de sesión</i>	170
<i>Caso de uso 9: bitácora de inventarios</i>	171
<i>Caso de uso 10: bitácora de expedientes</i>	173
<i>Caso de uso 11: bitácora de proveedores</i>	174
<i>Caso de uso 11: panel de control</i>	175
DIAGRAMAS DE FLUJO	176
<i>Panel de control</i>	177
<i>Tipo de rol</i>	178
<i>Registro usuario</i>	179
<i>Reportes de Auditoría</i>	180
<i>Expedientes de soporte</i>	181
<i>Mantenimiento</i>	183
DIAGRAMAS DE SECUENCIA	184
MODELO ENTIDAD-RELACIÓN.....	190
DICCIONARIO DE DATOS	193
DESARROLLO DEL SISTEMA	197
<i>Pantalla del Login</i>	197
<i>Pantalla de Panel de control</i>	198
<i>Pantalla para agregar usuarios</i>	199
<i>Pantalla reportes Auditoría</i>	202
<i>Pantalla de expedientes</i>	203
<i>Pantalla de monitoreo</i>	204
<i>Pantalla de movimientos</i>	205
<i>Pantalla de proveedores</i>	206

<i>Pantalla de inventarios</i>	208
REFERENCIAS	210
ANEXO 1. CONSTANCIA DE REVISIÓN FILOLÓGICA	219

Índice de figuras

FIGURA 1: CARACTERÍSTICAS DEL LENGUAJE C	14
FIGURA 2 HISTORIA DE LA PROGRAMACIÓN	16
FIGURA 3: PARADIGMAS DE LA PROGRAMACIÓN	17
FIGURA 4 : MAPA CONCEPTUAL DE FUNDAMENTOS DE PROGRAMACIÓN.....	19
FIGURA 5: LENGUAJE DE PROGRAMACIÓN	21
FIGURA 6: LENGUAJES DE ALTO NIVEL VERSUS BAJO NIVEL.....	22
FIGURA 7: LENGUAJES TIPADOS ESTÁTICOS	23
FIGURA 8: LENGUAJES TIPADOS DINÁMICOS	23
FIGURA 9: LENGUAJES DE PROGRAMACIÓN	24
FIGURA 10 : ¿QUÉ SON LOS PARADIGMAS DE PROGRAMACIÓN?	25
FIGURA 11 : PROGRAMACIÓN ORIENTADA A OBJETOS	26
FIGURA 12: PROGRAMACIÓN FUNCIONAL	27
FIGURA 13 : LOS ALGORITMOS EN LA PROGRAMACIÓN	28
FIGURA 14 : SISTEMA DE GESTIÓN DE ACTIVOS	30
FIGURA 15 : MÉTODO ABC.....	33
FIGURA 16 : MÉTODO PEPS (FIFO).....	34
FIGURA 17 : MÉTODO EOQ ADAPTADO	35
FIGURA 18 : CONTEO LIFO	36
FIGURA 19 : CONTEO CÍCLICO	37
FIGURA 20: SISTEMA DE INFORMACIÓN (SI)	38
FIGURA 21 : SISTEMAS DE PROCESAMIENTO DE TRANSACCIONES (TPS).....	39
FIGURA 22: SISTEMAS DE INFORMACIÓN GERENCIAL (MIS)	40
FIGURA 23 : SISTEMAS DE CONTROL DE PROCESOS DE NEGOCIO (BPM)	41
FIGURA 24 : SISTEMAS DE INFORMACIÓN DE MARKETING (SIM)	42
FIGURA 25 : SISTEMAS DE COLABORACIÓN EMPRESARIAL (ERP)	43
FIGURA 26 : SISTEMAS DE SOPORTE A LA TOMA DE DECISIONES (DSS)	44
FIGURA 27 : SISTEMAS DE INFORMACIÓN EJECUTIVA (EIS)	45
FIGURA 28: CICLO DE VIDA DEL SOFTWARE	47
FIGURA 29 : REQUISITOS DEL SISTEMA.....	51
FIGURA 30: COMPONENTES	59

FIGURA 31: DIAGRAMA UML.....	60
FIGURA 32 : DIAGRAMA CASO DE USO	62
FIGURA 33 : SQL SERVER	63
FIGURA 34: EJEMPLO DE UN CASO DE USO.....	66
FIGURA 35: HISTORIA DE LAS BASES DE DATOS.	67
FIGURA 36: BASE DE DATOS	68
FIGURA 37: BASE DE DATOS OPEN SOURCE	71
FIGURA 38: METODOLOGÍAS ÁGILES Y TRADICIONAL.....	75
FIGURA 39: EJEMPLO DE PANTALLAS DE VB.NET-FORMS	77
FIGURA 40:EJEMPLO DE PANTALLA VB.NET	78
FIGURA 41: DIAGRAMA DE CLASES DEL MODELO.....	79
FIGURA 42: DIAGRAMA DE CLASES DE LA GUI.....	80
FIGURA 43: LA GUI DE LA APLICACIÓN DE DEMOSTRACIÓN DE LA AGENDA TELEFÓNICA	82
FIGURA 44: FORMULARIO PARA AGREGAR UN NUEVO CONTACTO	83
FIGURA 45: DIAGRAMA DE ISHIKAWA.....	86
FIGURA 46 : MAPA SATELITAL DE INSTACREDIT S.A.....	89
FIGURA 47: ORGANIGRAMA DE INSTACREDIT S.A.	90
FIGURA 48: ORGANIGRAMA DE INSTACREDIT S.A.	91
FIGURA 49: MARCO METODOLÓGICO.....	97
FIGURA 50:ENFOQUE DE LA INVESTIGACIÓN	102
FIGURA 51: ENFOQUE DE INVESTIGACIÓN.....	103
FIGURA 52: ENFOQUE CUANTITATIVO.....	108
FIGURA 53: ENFOQUE CUALITATIVO	109
FIGURA 54 : FUENTES DE INFORMACIÓN	114
FIGURA 55: TIPOS DE VARIABLES, INDEPENDIENTES Y DEPENDIENTES	115
FIGURA 56 : DIAGRAMA DE FLUJO DE USO ADMINISTRADOR.	118
FIGURA 57 : DIAGRAMA DE FLUJO ESTÁNDAR.....	119
FIGURA 58: CASO DE USO.	120
FIGURA 59 : GESTIÓN DE RIESGO	122
FIGURA 60 : MATRIZ DE RIESGO	126
FIGURA 61 : MAPA DE CALOR DE RIESGO.....	127
FIGURA 62 : RESULTADOS DE MAPA DE CALOR	128
FIGURA 63:FACTIBILIDAD DE SISTEMAS	129
FIGURA 64: ESTUDIO DE FACTIBILIDAD.....	137
FIGURA 65: CRONOGRAMA DE TAREAS Y CONTROL MÉTODO KANBAN.....	140

FIGURA 66: DIAGRAMA GENERAL DE CASOS DE USO	161
FIGURA 67: DIAGRAMA DE CASO DE USO - LOGIN	162
FIGURA 68 : DIAGRAMA DE CASO DE USO DE REGISTRO DE USUARIOS.....	163
FIGURA 69: DIAGRAMA DE CASO DE USO DE REGISTRO DE ACTIVOS	164
FIGURA 70: DIAGRAMA DE CASO DE USO DE REGISTRO DE PROVEEDORES.....	165
FIGURA 71: DIAGRAMA DE CASO DE USO DE BOLETAS DE MOVIMIENTO	166
FIGURA 72: DIAGRAMA DE CASO DE USO DE EXPEDIENTES	168
FIGURA 73: DIAGRAMA DE CASO DE USO DE AUDITORÍA	169
FIGURA 74: DIAGRAMA DE CASO DE USO BITÁCORA INICIO DE SESIÓN	170
FIGURA 75: BITÁCORA DE INVENTARIOS	171
FIGURA 76: BITÁCORA DE EXPEDIENTES	173
FIGURA 77: BITÁCORA DE PROVEEDORES	174
FIGURA 78: BITÁCORA DE PROVEEDORES	175
FIGURA 79: DIAGRAMA DE FLUJO LOGIN.....	176
FIGURA 80: DIAGRAMA DE FLUJO PANEL DE CONTROL	177
FIGURA 81: DIAGRAMA DE FLUJO TIPO DE ROL.....	178
FIGURA 82: DIAGRAMA DE FLUJO REGISTRO USUARIO	179
FIGURA 83: DIAGRAMA DE FLUJO REPORTES DE AUDITORÍA.....	180
FIGURA 84: DIAGRAMA DE FLUJO REGISTRO EXPEDIENTES	181
FIGURA 85: DIAGRAMA DE FLUJO REGISTRO MOVIMIENTO DE BOLETAS.....	182
FIGURA 86: DIAGRAMA DE FLUJO MANTENIMIENTO	183
FIGURA 87: DIAGRAMA DE SECUENCIA LOGIN	184
FIGURA 88: DIAGRAMA DE SECUENCIA PRIMER INICIO DE SESIÓN	184
FIGURA 89: DIAGRAMA DE SECUENCIA DE OLVIDO SU CONTRASEÑA.....	185
FIGURA 90: DIAGRAMA DE SECUENCIA PANEL DE CONTROL	185
FIGURA 91: DIAGRAMA DE SECUENCIA AGREGAR USUARIO.....	186
FIGURA 92: DIAGRAMA DE SECUENCIA AUDITORÍA.....	186
FIGURA 93: DIAGRAMA DE SECUENCIA EXPEDIENTES	187
FIGURA 94: DIAGRAMA DE SECUENCIA MOVIMIENTO DE BOLETAS	187
FIGURA 95: DIAGRAMA DE SECUENCIA DE PROVEEDORES	188
FIGURA 96: DIAGRAMA DE SECUENCIA PARA AGREGAR NUEVO PROVEEDOR	188
FIGURA 97: DIAGRAMA DE SECUENCIA DE INVENTARIÓ	189
FIGURA 98: MODELO RELACIONAL.....	190
FIGURA 99:MODELO RELACIONAL.....	191
FIGURA 100: MODELO RELACIONAL.....	192

FIGURA 101: LOGIN.....	197
FIGURA 102: PANEL DE CONTROL.....	198
FIGURA 103: AGREGAR USUARIOS	199
FIGURA 104: EJEMPLO DE CÓMO SE AGREGA UN USUARIO.....	200
FIGURA 105: EJEMPLO DE CÓMO BUSCAR USUARIOS.	201
FIGURA 106: PANTALLA DE REPORTES PARA AUDITORÍA.	202
FIGURA 107: PANTALLA EXPEDIENTES	203
FIGURA 108: PANTALLA DE MONITOREO.....	204
FIGURA 109: PANTALLA DE MOVIMIENTOS.....	205
FIGURA 110: PANTALLA PROVEEDORES.....	206
FIGURA 111: PANTALLA PARA AGREGAR NUEVOS PROVEEDORES.....	207
FIGURA 112: PANTALLA DE INVENTARIOS.....	208
FIGURA 113: OPCIÓN DE SUBIR ARCHIVOS CSV PARA ACTUALIZAR INVENTARIOS.....	209

Índice de tablas

TABLA 1: CANTIDAD DE EMPLEADOS POR ÁREA.....	92
TABLA 2 : FACTIBILIDAD TÉCNICA	130
TABLA 3: FACTIBILIDAD ECONÓMICA	131
TABLA 4 : COSTO DEL PROYECTO.....	132
TABLA 5: COSTO DEL PROYECTO	133
TABLA 6 : FACTIBILIDAD OPERATIVA	137
TABLA 7 : REQUERIMIENTO DE USUARIO RUM01.....	146
TABLA 8: REQUERIMIENTO DE USUARIO RUM02	147
TABLA 9: REQUERIMIENTO DE USUARIO RUM03	147
TABLA 10: REQUERIMIENTO DE USUARIO RUM04	148
TABLA 11: REQUERIMIENTO DE USUARIO RUM05	149
TABLA 12: REQUERIMIENTO DE USUARIO RUM06	149
TABLA 13: REQUERIMIENTO DE USUARIO RUM07	150
TABLA 14: REQUERIMIENTO DE USUARIO RUM08	151
TABLA 15: REQUERIMIENTO DE USUARIO RUM09	151
TABLA 16: REQUERIMIENTO DE USUARIO RUM10	152
TABLA 17: REQUERIMIENTO DE USUARIO RUM11	152
TABLA 18: REQUERIMIENTO DE USUARIO RUM12	153
TABLA 19: REQUERIMIENTO FUNCIONAL RFM01	154
TABLA 20: REQUERIMIENTO FUNCIONAL RFM02	154

TABLA 21: REQUERIMIENTO FUNCIONAL RFM03	155
TABLA 22: REQUERIMIENTO FUNCIONAL RFM04	155
TABLA 23: REQUERIMIENTO FUNCIONAL RFM22	156
TABLA 24: REQUERIMIENTO FUNCIONAL RFM06	156
TABLA 25: REQUERIMIENTO FUNCIONAL RFM07	157
TABLA 26: REQUERIMIENTO FUNCIONAL RFM08	157
TABLA 27: REQUERIMIENTO FUNCIONAL RFM09	158
TABLA 28: REQUERIMIENTO NO FUNCIONAL RNFM01	158
TABLA 29: REQUERIMIENTO NO FUNCIONAL RNFM02	159
TABLA 30: REQUERIMIENTO NO FUNCIONAL RNFM03	159
TABLA 31: REQUERIMIENTO NO FUNCIONAL RNFM04	160
TABLA 32: REQUERIMIENTO NO FUNCIONAL RNFM05	160
TABLA 33: LOGIN	162
TABLA 34: REGISTRO DE USUARIO	163
TABLA 35: REGISTRO DE ACTIVOS	164
TABLA 36: REGISTRO PROVEEDORES	166
TABLA 37: REGISTRO DE BOLETAS DE MOVIMIENTO	167
TABLA 38: REGISTRO DE EXPEDIENTES	168
TABLA 39: AUDITORÍA	169
TABLA 40: BITÁCORA DE SESIONES	171
TABLA 41: BITÁCORA DE INVENTARIOS	172
TABLA 42: BITÁCORA DE EXPEDIENTES	173
TABLA 43: PANEL DE CONTROL	175
TABLA 44: TABLA USUARIOS	193
TABLA 45: TABLA INVENTARIO	193
TABLA 46: TABLA MOVIMIENTOS	194
TABLA 47: TABLA CONTRATOS	194
TABLA 48: TABLA EXPEDIENTES	194
TABLA 49: TABLA PROVEEDORES	195
TABLA 50: TABLA BITÁCORA INVENTARIO	195
TABLA 51: TABLA BITÁCORA CONTRATOS	195
TABLA 52: TABLA BITÁCORA	195
TABLA 53: TABLA BITÁCORA CONEXIONES	196

Dedicatoria y agradecimiento

Dedico este trabajo a las personas más importantes para mí; a mi pareja Karen Vetrani Chavarría, mi papá Bismar Argüello Ruiz y a mi angelito, mi bebé desde el cielo que me acompaña y me ayuda a ser mejor persona.

Ustedes siempre me impulsan hacia adelante, motivándome a ser mejor que ayer, gracias por estar a mi lado en esas largas noches de desvelo, mientras terminaba trabajos de investigación o luchaba con la programación que no funcionaba bien al inicio. Gracias por acompañarme en todo este proceso de tesis y, sobre todo, por siempre creer en mí y no dejar que me rindiera con mi sueño.

No tengo palabras para describir todo lo que siento al escribir esto. Solo Dios sabe el esfuerzo que he hecho para llegar hasta aquí y culminar una etapa en mi vida, aunque sé que no será la última.

Resumen

Esta tesis expone el desarrollo de un sistema digital denominado ASSETPRO, cuya finalidad es optimizar la gestión de activos en el área de soporte técnico de Instacredit S.A. Se plantea, desde una perspectiva integral, la necesidad de sustituir procesos manuales como la asignación de equipos, el manejo de inventarios mediante hojas de Excel y el almacenamiento físico de documentos, por una solución tecnológica que garantice mayor precisión y trazabilidad.

El proyecto propone la automatización de tareas claves como digitalizar boletas de entrega y retiro (con envío por correo), actualizar el inventario en tiempo real, al generar la boleta de movimiento y crear expedientes digitales para el seguimiento de incidencias; además de incorporar un sistema de gestión de usuarios con roles específicos, que facilite también la administración de proveedores y la emisión de reportes de auditoría. Una solución que, en esencia, busca transformar la operatividad interna, reduciendo errores y duplicidades.

Basado en antecedentes nacionales e internacionales, como modelos fundamentados en COBIT, ITIL e ISO55000, la investigación se sustenta en un riguroso análisis de requerimientos tanto funcionales como no funcionales y en el diseño de una arquitectura adaptable y escalable. Así mismo, se emplean metodologías de desarrollo tradicionales y ágiles que aseguran la integración exitosa del sistema con las tecnologías preexistentes en la empresa.

En resumen, la tesis no solo presenta una propuesta técnica viable y económicamente factible, sino que también destaca la importancia de una planificación estratégica y del compromiso institucional para impulsar la digitalización y la eficiencia operativa. Así, mediante la implementación de ASSETPRO, se aspira a mejorar la productividad y la seguridad en la

gestión de activos digitales, marcando un hito hacia la modernización del área de soporte técnico y el fortalecimiento de la cultura de inversión en Instacredit S.A.

Abstract

The thesis presents the development of a digital system called ASSETPRO, whose purpose is to optimize asset management in the technical support area of Instacredit S.A. It raises from a comprehensive perspective the need to replace manual processes such as the assignment of equipment, inventory management through Excel sheets and physical storage of documents with a technological solution that guarantees greater precision and traceability.

The project proposes the automation of key tasks: digitizing delivery and withdrawal slips (with mail delivery), updating the inventory in real time when generating the movement slip and generating digital files for the monitoring of incidents, in addition to incorporating a user management system with specific roles, which also facilitates the administration of suppliers and the issuance of audit reports. A solution that, in essence, seeks to transform internal operations, reducing errors and duplications. Based on national and international backgrounds such as COBIT, ITIL and ISO55000 fundamental models, the research is supported by a rigorous analysis of both functional and non-functional requirements and the design of an adaptable and scalable architecture. Traditional and agile development methodologies are also used to ensure the successful integration of the system with the pre-existing technologies in the company.

In summary, the thesis not only presents a viable and economically feasible technical proposal, but also highlights the importance of strategic planning and institutional commitment to promote digitalization and operational efficiency. Thus, through the implementation of ASSETPRO, it is hoped to improve productivity and security in the management of digital

assets, marking a milestone towards the modernization of the technical support area and strengthening the invasion culture at Instacredit S.A.

CAPÍTULO I. PROBLEMA

Planteamiento del problema

En la actualidad, la empresa Instacredit S.A. cuenta con varias sedes en Costa Rica, con sus oficinas centrales ubicadas en San José, cantón Central, distrito Catedral. En estos momentos, enfrenta diversos problemas relacionados con la desactualización de ciertos procesos que continúan realizándose de forma manual. Entre los cuales se incluyen la asignación de equipos, la gestión de boletas de entrega y retiro de activos, el manejo de inventarios mediante hojas de Excel y la conservación de documentos en formato físico.

El almacenamiento de documentos físicos en archivos y bodegas ha generado inconvenientes como la pérdida de documentos debido a daños, extravío o el paso del tiempo. Asimismo, la duplicación de activos es un problema recurrente, ya que, al no contar con un sistema adecuado para su asignación, estos son ingresados manualmente en hojas de Excel utilizadas como inventario. Además, la falta de información estructurada y la limitada trazabilidad dificultan la presentación de informes en auditorías, lo que impacta negativamente la eficiencia del equipo de soporte técnico.

A partir de lo anterior, esta investigación plantea una solución práctica y comprobada que aborda problemas específicos. La propuesta incluye la digitalización de boletas de entrega y retiro de equipos, permitiendo su envío por correo y la actualización automática del inventario. Asimismo, se contempla la implementación de un sistema de inventarios con funcionalidades para el ingreso, actualización, inhabilitación y búsqueda de activos. También se prevé la creación de expedientes digitales para el seguimiento de amonestaciones dirigidas al equipo de soporte en casos de incumplimiento laboral o actitudes inadecuadas.

Adicionalmente, el sistema incluye la gestión de usuarios con roles específicos (ingresar, inhabilitar, actualizar y buscar información). Se incorpora una sección dedicada a los

proveedores, con el objetivo de ofrecer mayor visibilidad sobre los contratos vigentes y aquellos próximos a vencer. Además, se habilita una opción para descargar o enviar archivos a los auditores en el momento en que sean requeridos.

Con la implementación de estos cambios, se busca garantizar una actualización y trazabilidad del 100 % de los activos de la empresa y de la información relevante para su gestión. En conclusión, la investigación presenta una solución viable para optimizar la gestión del equipo de soporte, asegurando un incremento en la eficiencia operativa y en la organización de los procesos internos.

Objetivo

Objetivo general

Implementar un sistema de gestión de activos digitales en el área de soporte técnico de Instacredit S.A. para manejar todo de manera digital, automatizar procesos y tener un mayor control de la trazabilidad y optimización de tiempo, a fin de hacer al departamento más eficiente.

Objetivos específicos

1. Evaluar el estado actual de la gestión de documentos y activos en el área de soporte, identificando las ineficiencias y problemas derivados del uso de documentos físicos y procesos manuales.

2. Definir los requerimientos técnicos y operativos necesarios para implementar el sistema de gestión de activos que permita reducir la duplicación de activos, pérdida de documentos y optimizar el tiempo operativo.
3. Diseñar los procesos del sistema utilizando notación UML, para automatizar los procesos e integrar el sistema de gestión de activos con las plataformas y tecnologías existentes en la empresa.
4. Programar el sistema de gestión de activos e integrarlo con las plataformas tecnológicas existentes en la empresa, así como realizar pruebas del sistema para garantizar la funcionalidad de los requerimientos.

Justificación

La investigación se centra en las malas prácticas implementadas en la empresa, al llevar documentos en papel y utilizando herramientas incorrectas para los procesos que se realizan actualmente, lo que dificulta las labores. La falta de un sistema centralizado adecuado ha tenido un efecto negativo en las eficiencias, provocando inexactitudes en las auditorías realizadas al Departamento de Soporte. Además, el faltante de los datos actualizados en el momento que se requieren muestra la importancia de contar con un sistema de gestión de activos.

El análisis previo propone una alternativa para abordar estos inconvenientes, con el objetivo de mejorar la gestión de activos y de la información de la empresa Instacredit. Esto involucra actualizar los inventarios en el momento que se genera una boleta (también se pueden agregar productos de forma manual, por medio de un CSV cargándolo al sistema que actualizará todas las líneas que se ingresan), boletas electrónicas entregadas por correo, seguimiento de

expedientes digitales, reportes de auditorías con los datos actualizados al día, mejor control de los contratos con proveedores con alertas tempranas de vencimientos, control de eventos de la aplicación, como estado de salud de las tablas y latencias entre la aplicación y la base de datos, así como control de usuarios con su respectivo rol.

Esto ayudará a incrementar la productividad, trazabilidad y transparencia en los procesos. De esta manera, la observación resalta la relevancia significativa en la comunidad, al implementar un sistema tecnológico que fomenta la exactitud y la constante actualización de la información, marcando por mucho la cultura de la empresa del círculo virtuoso de cooperación. Al implementar estos cambios, que son de peso, aumentaría la eficiencia del departamento de soporte porque, al no tener que hacer los procesos manuales, se ahorraría tiempo que se puede invertir en otras actividades de valor para la empresa; la cual es una de las metas propuestas en esta investigación. A su vez, se incrementaría la precisión en el manejo de inventarios (entregas, retiros, bajas, donaciones, robo, etc.). Por último, estas mejoras ayudarán a diferentes áreas, ya que se les podrá facilitar la información con mayor eficiencia, por ejemplo, en el área de auditoría, compras y recursos humanos.

Antecedentes

Nacionales

Como parte de los antecedentes de este proyecto, el primer trabajo en estudio se titula: *Modelo de gestión de activos para el área de tecnologías de la información y comunicación de una empresa de servicios de e-learning*, presentado por Aguilar Barboza (2024). Este proyecto

se basa en COBIT 5 y la guía de prácticas de ITIL, con el propósito de mejorar los procesos en la gestión de activos y aumentar la competitividad de la empresa mediante la implementación de buenas prácticas internacionales.

El estudio identificó que el proceso de gestión de activos TIC en la empresa era informal y subdesarrollado. Como resultado, Aguilar Barboza (2024) propuso una guía para mejorar la toma de decisiones y obtener ventajas competitivas al implementar un modelo que integra los marcos mencionados. Además, el uso de COBIT 5 e ITIL 5 garantiza una gestión más eficiente y estructurada de activos TIC en la organización. Este trabajo aporta una base sólida con buenas prácticas para la gestión de activos, colaborando en la estructura que se implementará en el sistema de gestión de activos.

En el segundo trabajo titulado: *Gestión digital de expedientes en Sysco*, Salas-Vargas (2023) se enfocó en la digitalización de los expedientes físicos del área de recursos humanos de la empresa Sysco. El objetivo principal fue desarrollar una aplicación utilizando REACT, para gestionar los expedientes de manera más eficiente y segura.

El sistema desarrollado por Salas Vargas (2023) facilita la digitalización de los expedientes de los colaboradores, lo que permite un acceso remoto más rápido y seguro a la información. Además, reduce los riesgos de pérdida o daño de los documentos físicos y mejora la toma de decisiones en la empresa, promoviendo la transformación digital en la gestión de recursos humanos.

Este trabajo proporciona una idea clara sobre el uso de la tecnología como REACT para digitalización y gestión de documentos; lo cual proporciona ideas valiosas para el sistema que se desarrollará para la gestión de expedientes y boletas de entrada, retiro, etc.

En el tercer trabajo titulado: *Diseño de Propuesta de Modelo de Gestión de Activos basado en la Norma ISO 55000 y un Sistema Integrado de Gestión del Espacio de Trabajo (IWMS)*, Campos Zuñiga (2014) propone un modelo de gestión de activos basado en la norma internacional ISO 55000. Este sistema tiene como objetivo “mejorar la gestión total de los activos físicos de una entidad durante todo su ciclo vital”.

El proyecto tiene como importancia la planificación de estrategia y el liderazgo en la implementación de un sistema de gestión de activos, así como la logística de los recursos, incluyendo colaboradores, económicos, activos fijos y los intangibles. Por lo tanto, este sistema ayudará a integrar la gestión de activos, reducir costos, riesgos y mejorar la eficiencia del área de soporte. También mostrará la importancia de la planificación para alcanzar las metas.

Internacionales

En el ámbito internacional, se considera el proyecto titulado: *Análisis del uso de herramientas informáticas para la gestión de activos empresariales en las empresas del sector alimenticio registradas en la Cámara de la Pequeña y Mediana Empresa de Pichincha*, presentado por Chávez Liquinchana (2020). El estudio reveló que las empresas del sector alimenticio tienen un conocimiento básico en la gestión de activos. Por lo que se propone implementar un sistema de gestión de activos (EAM) que incluya estrategias de mantenimiento,

evaluación de opciones y una gerencia efectiva de la información, con el fin de maximizar la rentabilidad de los activos. Este análisis puede ser de utilidad para el presente proyecto, ya que muestra un marco para la implementación de un sistema de gestión de activos que mejora la eficiencia operativa.

En el segundo trabajo titulado: *Sistema para el control de activos informáticos de la facultad 4*, Herrera Vive (2015) define el objetivo principal de integrar el proceso de inventarios de los activos de *hardware*, el cual anteriormente se realizaba de forma manual, lo que generaba duplicación de datos, pérdida de información y errores en el registro de activos.

Este trabajo es de utilidad, ya que representa una guía para la informatización y automatización del inventario que se aborda en el proyecto, mejorando la eficiencia y precisión en la gestión de activos.

Finalmente, en el tercer trabajo titulado: *Análisis, Diseño y Construcción de una Aplicación Web para la Gestión de Activos de TI*, Portelles Cobas (2015) parte de la necesidad de la empresa de contar con una solución tecnológica que centralice la información de los activos, permitiendo una mejor trazabilidad y facilitando la generación de reportes.

El desarrollo se basa en metodología ágil y herramientas tecnológicas como Microsoft Visual Studio, SQL Server y Bootstrap. Además, la solución propuesta incluye funcionalidades como creación, asignación, eliminación y reporte de activos de TI, con un fuerte enfoque en la mejora del control y la reducción de los errores en la gestión manual.

Este proyecto es de gran utilidad para el trabajo por presentar, ya que proporciona las bases sólidas para el desarrollo de la aplicación de gestión de activos, debido a que implementará metodologías ágiles y se quiere optimizar los tiempos.

Proyecciones

Alcances

1. Mejorar la gestión y control de los inventarios en la empresa Instacredit S.A., para evitar la pérdida de documentos, duplicación de información, reducir los tiempos de búsqueda de la información y mostrar una trazabilidad en tiempo real.
2. Desarrollar un sistema de gestión de activos digitales para la empresa Instacredit S.A. que cumpla con los requisitos de seguridad, que se pueda inhabilitar, editar, agregar y subir CSV.
3. Desarrollar un Login para ingreso a la aplicación con su usuario debidamente creado por el administrador del sistema y con sus roles correspondientes.
4. Desarrollar módulos para proveedores, expedientes, auditoría, usuarios, inventarios y monitoreo. Además, incluir una clase para los permisos según el rol que se tenga por usuario.
5. No se implementará un manual de usuario.
6. Brindar capacitación y acompañamiento del proceso al personal de soporte de Instacredit S.A.
7. Implementar los permisos que sean requeridos por el usuario.

8. No se realizarán estudios de factibilidad comercial, de tiempo, política o legal.

Limitaciones

1. Resistencia al cambio: algunos de los colaboradores pueden mostrar resistencia a abandonar los métodos tradicionales, lo que puede retrasar el tiempo de adaptación para el sistema.
2. Migración de los datos: debido a que no se cuenta con un sistema previo de inventario, todos los equipos se manejaban en un Excel, por lo que la información se inserta masivamente sin depuración, para mantener la continuidad y cumplir con las auditorías, inhabilitando equipos que ya no están disponibles (bajas, robos, donaciones) para evitar errores en las boletas.
3. Integración de API: para acceder al padrón electoral, se trabajará con un archivo txt con información del padrón, debido a la falta de un API gratuito que se pueda adaptar al proyecto. La integración se resuelve luego de presentar la tesis, ya que la empresa cuenta con una y no se requiere comprar o invertir.
4. Gastos asociados a licencias y *hardware*: el proyecto no contempla los gastos asociados a licencias de *software* ni a equipo *hardware* requerido para su implementación, ya que la empresa cuenta con la infraestructura tecnológica y con las licencias necesarias.
5. Servidor de envío de correos: no se cuenta con acceso al servidor de envío de correos de la empresa durante la fase de desarrollo. Se utilizará

temporalmente un servidor SMTP de Gmail para el envío de boletas. Una vez se encuentre en la fase de implementación, se utilizará la infraestructura existente de la empresa para asegurar la seguridad de la información.

6. Capacitación del personal: la capacitación será impartida por el desarrollador, pero se depende del equipo de capacitación, quienes brindarán asistencia en este proceso, por lo que se dependerá de la agenda de ellos.

CAPÍTULO II. MARCO TEÓRICO

Historia de la programación

En los últimos años, los lenguajes de programación han ido evolucionando en el desarrollo de sistemas o *software*, con el objetivo principal de facilitar al usuario las actividades que realiza día con día. Por tal motivo, como programador, es importante conocer los conceptos básicos de programación, los tipos de lenguajes que se utilizan para el desarrollo y su funcionamiento en la interpretación de algoritmos, así como para dar solución a los problemas que pudieran presentarse (Lenguajes de Programación|Monterde y Marín, s.f.).

Definición

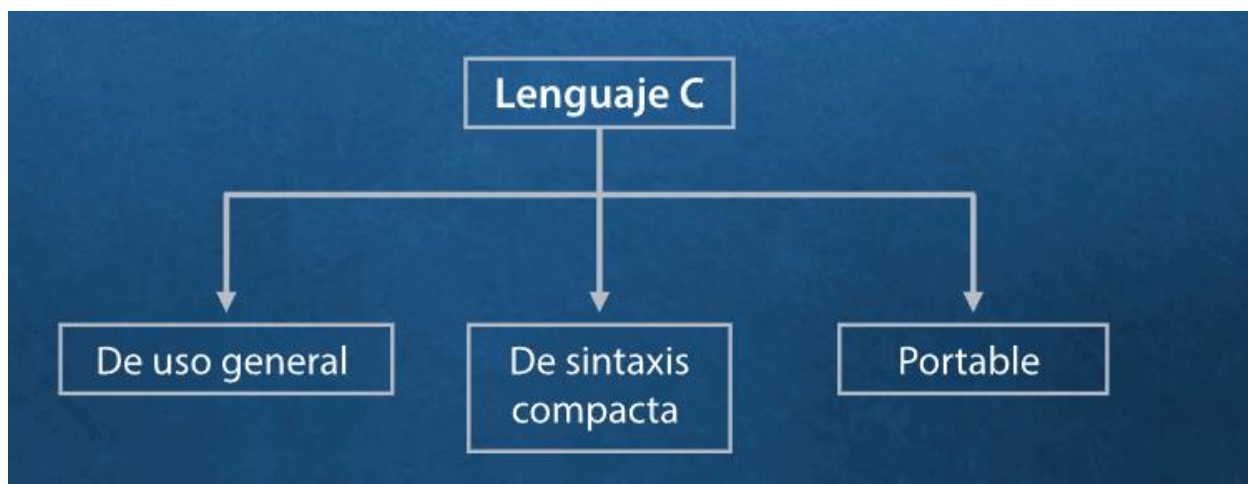
En términos generales, un lenguaje de programación es una herramienta que permite desarrollar *software* o programas para computadora. Los lenguajes de programación son empleados para diseñar e implementar programas encargados de definir y administrar el comportamiento de los dispositivos físicos y lógicos de una computadora. Lo anterior se logra mediante la creación e implementación de algoritmos de precisión que se utilizan como una forma de comunicación humana con la computadora.

A grandes rasgos, un lenguaje de programación se conforma de una serie de símbolos y reglas de sintaxis y semántica que definen la estructura principal del lenguaje y le dan un significado a sus elementos y expresiones. Por su parte, la programación es el proceso de análisis, diseño, implementación, prueba y depuración de un algoritmo, a partir de un lenguaje que compila y genera un código fuente ejecutado en la computadora.

La función principal de los lenguajes de programación es escribir programas que permiten la

comunicación usuario-máquina. Unos programas especiales (compiladores o intérpretes) convierten las instrucciones escritas en código fuente, en instrucciones escritas en lenguaje máquina (0 y 1) (Lenguajes de Programación|Monterde y Marín, s.f.).

Figura 1: Características del lenguaje C



Fuente: (Lenguajes de Programación|Monterde y Marín, s.f.).

Antecedentes

Profesor de matemáticas e inventor en la Universidad de Cambridge, Inglaterra, a mediados del siglo XIX, Charles Babbage fue el primero en concebir la idea de un lenguaje de programación, al predecir varias de las teorías en las que se basan las computadoras actuales.

Babbage desarrolló la idea de una máquina analítica programable que, por limitaciones tecnológicas de su época, no pudo ser construida. Junto con él, su colaboradora, Ada Lovelace, es considerada como la primera programadora de la historia, ya que escribió los primeros programas para la máquina concebida por Babbage en tarjetas perforadas, siguiendo una lógica

de programación muy similar a la empleada en la actualidad. Sin embargo, estos programas nunca pudieron verse ejecutados debido a que la máquina no fue construida.

Las técnicas empleadas por Babbage y Ada fueron seguidas por los primeros programadores de computadoras, quienes se valieron de tarjetas perforadas para introducir sus programas en estas.

En 1823, con el apoyo del gobierno británico, se aprobó el proyecto de construcción de una máquina de diferencias. Esta era un dispositivo mecánico diseñado para realizar sumas de forma repetitiva. Babbage abandonó el proyecto para dedicarse a su máquina analítica, influenciado por la creación de un fabricante de telas francés, Joseph Marie Jacquard, que había desarrollado una máquina tejedora con la capacidad de reproducir patrones de tejidos, leyendo información codificada en tarjetas perforadas de papel rígido.

Desde entonces, Babbage se propuso construir una máquina que efectuara cálculos matemáticos de precisión, empleando 20 dígitos, y que pudiera ser programada mediante tarjetas perforadas. Aun cuando esta idea quedó solo en el proyecto, fue una contribución muy importante para el diseño y funcionamiento de las computadoras actuales.

Charles Babbage es considerado el padre de la informática. A pesar de que su máquina nunca pudo ser desarrollada, sus ideas y diseños sirvieron para la construcción y el progreso de las primeras computadoras modernas.

Cuando surgió la primera computadora, la ENIAC (Electronic Numerical Integrator And Calculator), su programación se basaba en componentes físicos, o sea, se programaba invirtiendo directamente el *hardware* de la máquina: se cambiaban de sitio los cables para conseguir así la programación. Además, la entrada y salida de datos se realizaba mediante tarjetas perforadas.

Para crear un lenguaje de programación, es necesaria una herramienta que lo traduzca. Por lo que se describe, a continuación, cómo ha ido evolucionando esta herramienta en los últimos 50 años (Lenguajes de Programación|Monterde y Marín, s.f.).

Figura 2 Historia de la programación



Fuente: (Dos siglos de historia del lenguaje de programación|RedaccionONOFF, 2022).

Para las décadas de 1960 y 1970, se definieron varios paradigmas:

Simula: pionero en la orientación a objetos.

NORTH: orientado a la programación concatenativa.

C: fundamental para la programación de sistemas.

Prolog: introdujo la programación lógica.

ML: representó la programación funcional.

La mayoría de los lenguajes modernos descienden de estos paradigmas.

Figura 3: Paradigmas de la programación



Fuente: (¿Qué son los paradigmas de programación?).Ed.team., s.f.).

Década de 1980:

ADA: destinada a sistemas críticos y embebidos.

ML: importante en el aprendizaje automático de datos.

C++: extensión de C con orientación a objetos y alto rendimiento.

Década de 1990:

La llegada de la web impulsó la creación de lenguajes como los siguientes:

Perl: muy utilizado para manipulación de texto y procesamiento de archivos.

Python: reconocido por su legibilidad y sencillez.

Java: introdujo la idea de “escribir una vez y ejecutar en cualquier lugar” mediante la máquina virtual.

Ruby: orientado a objetos con un enfoque sencillo y flexible.

Además, surgieron lenguajes de secuencias para la web, como JavaScript, orientados a la ejecución en navegadores.

Desde el año 2000:

Los lenguajes se han centrado en la seguridad y la eficiencia:

- **Rust:** hace énfasis en la seguridad de la memoria.
- **Go:** facilita el desarrollo concurrente y confiable.

- **Swift:** renovó el desarrollo en iOS y macOS con un enfoque seguro y eficiente.

Actualmente, los lenguajes más demandados incluyen JavaScript, Python, Java, C# y SQL, abarcando campos tan diversos como la web, el desarrollo móvil, la inteligencia artificial y el análisis de datos.

Fundamentos de la programación

Figura 4 : Mapa conceptual de fundamentos de programación



Fuente: (MAPA CONCEPTUAL - GESTIÓN DE COMUNICACIÓN EN LOS PROYECTOS | Castellanos, s.f.).

¿Qué es un lenguaje de programación?

Un lenguaje de programación es un sistema de signos y reglas que permite a las personas comunicarse con las computadoras para indicarles qué operaciones deben ejecutar.

Tipos de programación

Según Tiffin University, cada tipo de programación tiene características particulares y métodos de aproximación distintos. Aquí se mencionan los principales:

Imperativa: se basa en una serie de instrucciones paso a paso que modifican el estado del programa.

Declarativa: describe el resultado deseado en lugar de detallar los pasos necesarios para lograrlo.

Funcional: considera la computación como la evaluación de funciones matemáticas, reduciendo la modificación de estados.

Lógica: define hechos y reglas, usando un motor de inferencia para deducir conclusiones.

Orientada a eventos: reacciona a sucesos o interacciones del usuario (por ejemplo, en interfaces gráficas o videojuegos).

Orientada a objetos: organiza el código en torno a “objetos” que agrupan datos y comportamientos relacionados.

Figura 5: Lenguaje de programación



Fuente: (Tipos y niveles de lenguajes de programación ejemplos |AAU, 2024).

Tipos de lenguajes de programación

De acuerdo con Latam, la elección de un lenguaje depende del tipo de proyecto y sus objetivos específicos. Cada uno ofrece características diferentes:

Bajo nivel: como ensamblador y lenguaje máquina. Se relacionan directamente con el *hardware* y ofrecen alto rendimiento, aunque son más complejos.

Alto nivel: ejemplos como Python, Java y C++. Ofrecen mayor legibilidad y productividad, al abstraer los detalles técnicos del *hardware*.

Figura 6: Lenguajes de alto nivel versus bajo nivel



Fuente: (Lenguajes de Alto nivel vs de Bajo nivel | EDteam, s.f.).

Tipado estático: lenguajes como Go, Java, C#, C++, TypeScript y Kotlin exigen declarar el tipo de la variable y no permiten cambiarlo posteriormente.

Tipado dinámico: Python, JavaScript, PHP y Ruby permiten cambiar el tipo de las variables durante la ejecución, aunque esto puede derivar en errores, si no se gestiona con cuidado.

Figura 7: Lenguajes tipados estáticos

```
var edad int = 10
fmt.Println(edad) // 10

edad = "EDcamp Bogotá 7.0"
fmt.Println(edad) // Error!
return go(f, seed, [])
}
```

Fuente: (¿Qué son los lenguajes tipados y no tipados? (Explicación sencilla)|Ed.team, s.f.).

Figura 8: Lenguajes tipados dinámicos

```
edad = 7
print(edad) # 7

edad = "EDcamp Bogotá 2023"
print(edad) # "EDcamp Bogotá 2023" No da error
```

Fuente: (¿Qué son los lenguajes tipados y no tipados? (Explicación sencilla)|Ed.team, s.f.).

Compilado: transforma el código fuente a código máquina antes de ejecutarse, ofreciendo un rendimiento más alto. Ejemplos: C#, C++ y Go.

Interpretado: se traduce y ejecuta en tiempo real, como en JavaScript, PHP, Python y Ruby, lo que puede ralentizar su rendimiento.

Intermedio: utiliza “Bytecode” (como Java y Kotlin) para lograr portabilidad entre distintos sistemas operativos.

Figura 9: Lenguajes de programación



Fuente: (Características del lenguaje de programación|ISC, s.f.).

Paradigmas de programación

Según Robert C. Martin, un paradigma representa principios y guías para estructurar el código. Los paradigmas imperativos (incluyendo orientación a objetos) y declarativo (incluyendo funcional y lógico) influyen en la forma de pensar de los desarrolladores.

Figura 10 : ¿Qué son los paradigmas de programación?



Fuente: (¿Qué son los paradigmas de programación? | EDteam, s/f).

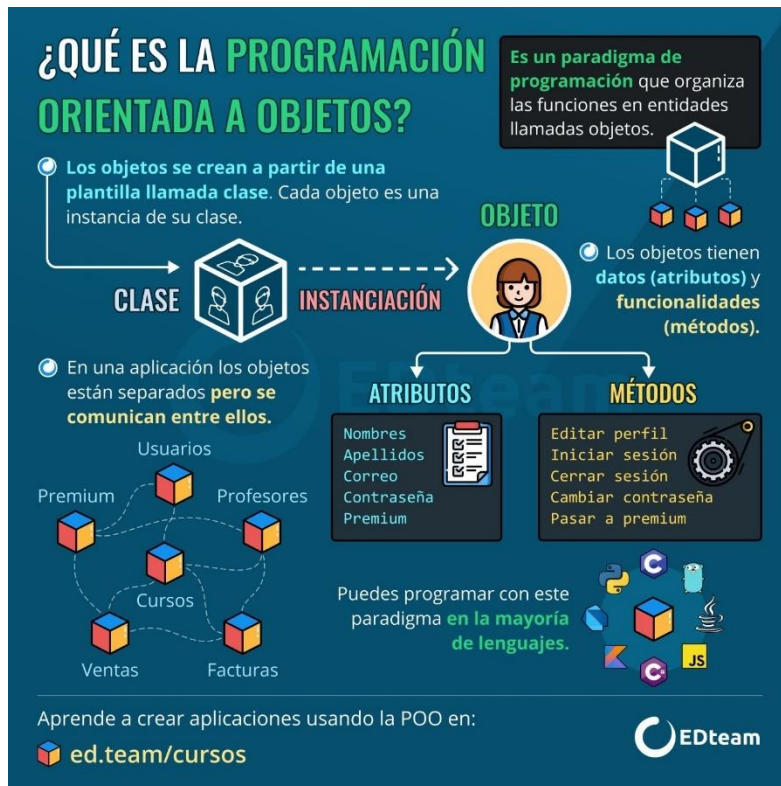
Orientada a objetos: se basa en la creación de clases que agrupan funciones y datos, lo que favorece la reutilización y simplifica la creación de sistemas complejos.

Funcional: trata las funciones como valores que pueden pasarse como argumentos, con énfasis en la inmutabilidad y la ausencia de efectos secundarios.

Imperativa: especifica secuencias de pasos para cambiar el estado del sistema, lo que otorga control detallado, pero puede complicar el mantenimiento.

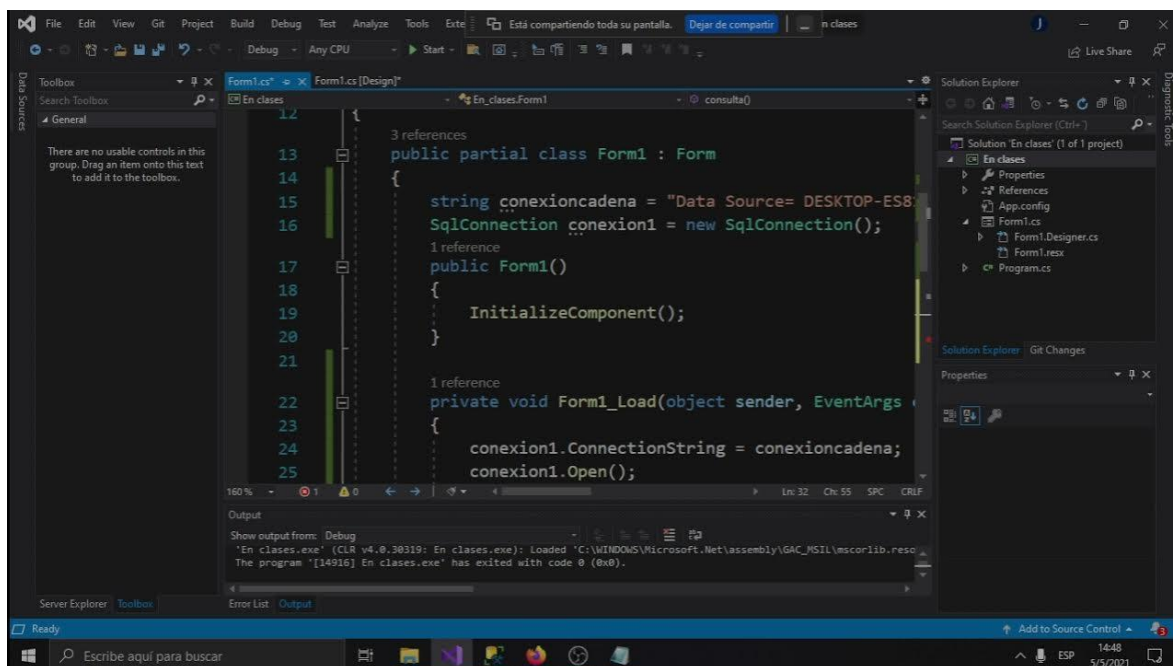
Declarativa: describe los resultados deseados y confía en reglas o lógica subyacentes para lograr el objetivo.

Figura 11 : Programación orientada a objetos



Fuente: (¿Qué es la Programación orientada a objetos?|Edteam, 2020).

Figura 12: Programación funcional



Fuente: (Crud con Programacion Funcional en Windows Forms C#|Zambrana, 2021).

Algoritmos de programación

Los algoritmos representan la base de cualquier *software*, ya que ayudan a resolver problemas de manera eficiente y guían la toma de decisiones en sistemas complejos.

Figura 13 : Los algoritmos en la programación



Fuente: (Algoritmos en la programación |Ed.team, 2021).

Sistema de gestión de activos digitales

Una solución de gestión de activos digitales es una solución de *software* y sistemas que proporciona un enfoque sistemático para almacenar, organizar, gestionar, recuperar y distribuir de manera eficiente los materiales digitales de una organización (IBM, ¿Qué es la gestión de activos digitales?, 2025). Este sistema integraría:

Registro y administración de usuarios: control de altas, bajas y roles de acceso, esto para poder determinar qué tantos permisos va a tener ese usuario, así como delimitar controles para mayor seguridad (IBM, Gestión de registros de usuarios, 2025).

Módulo de auditoría: permite generar y descargar reportes para verificar las operaciones realizadas, también permitiría generar reportes en el momento que sean necesarios, sin tener que

afectar la operativa de soporte técnico para buscar la documentación (IBM, Optimice las auditorías internas, 2025).

Expedientes de soporte: los expedientes de soporte son una herramienta fundamental para la gestión del equipo de soporte técnico dentro del sistema de gestión de activos digitales. Su propósito es centralizar y organizar toda la documentación relacionada con el personal de soporte, garantizando un seguimiento adecuado de su desempeño y procesos internos. Estos expedientes incluyen información clave como: seguimiento de desempeño, amonestaciones y medidas disciplinarias e incidencias internas (Glosario|{Docu Ware Europe GmbH}, 2025).

Control de boletas: se refiere a la gestión detallada de los movimientos de entrada y salida, con el objetivo de asegurar su trazabilidad y una administración eficiente. Según Tech-ID, en el ámbito logístico, la precisión y eficiencia en la gestión y trazabilidad de activos son fundamentales para el éxito operativo. La implementación del control de boletas sería trazabilidad, optimización de inventario y cumplimiento de la política interna de la empresa (Trazabilidad de Activos|Tech-ID, 2023).

Gestión de proveedores: la gestión de proveedores es el conjunto de procesos que permite a una empresa identificar, calificar, incorporar, realizar transacciones y colaborar con los proveedores adecuados para su negocio. Una gestión eficiente de proveedores facilita la comunicación y la negociación, permitiendo mantener un control más riguroso sobre los contratos establecidos. Además, implementar una gestión adecuada de proveedores ayuda a evitar situaciones similares, asegurando que los contratos sean competitivos y alineados con las necesidades de la organización (IBM, ¿Qué es la gestión de proveedores?, 2025).

Inventario de activos: monitoreo constante de la disponibilidad, ubicación y estado de los recursos digitales o físicos (Inventario de activos y gestión de la seguridad en SCI |Incibe.es, 2016). Tal como ocurre con los sistemas de inventario tradicionales, en el ámbito digital se requiere mantener un control riguroso de cada activo para garantizar su disponibilidad y proteger la información (ISO 55000|ISO, 2014).

Figura 14 : Sistema de gestión de activos



Fuente: (¿Qué son los Digital assets management (DAM) o sistema de gestión de activos digitales?|ESIC, 2019).

Métodos para el control de activos digitales

La gestión de activos digitales (DAM) puede referirse tanto a un proceso de negocio como a una forma de tecnología de gestión de la información o un sistema de gestión de activos digitales. La funcionalidad DAM ayuda a muchas organizaciones a crear un lugar centralizado donde pueden acceder a sus activos multimedia (IBM, ¿Qué es la gestión de activos digitales?, 2025).

Método ABC:

El origen del método ABC de gestión del *stock* proviene de la conocida como regla del 80/20 o principio de Pareto, según el que una pequeña parte del total de las cosas es la que contribuye a la mayor parte de la consecución de los resultados. Aplicando la regla 80/20 a la realidad de la empresa, un 20% del total de referencias son las que generarían el 80% de los beneficios (Método ABC de inventarios en almacén: que es|AR-Racking, s.f.).

Categoría A: según el sistema ABC, las referencias de la categoría A son las más importantes para la empresa. Son solo en torno a un 20% del inventario, pero suponen la mayoría del movimiento habitual de almacén, con mayor rotación, también los que aportan en torno al 80% de los ingresos de la empresa.

Al ser la categoría de referencias prioritaria, la empresa deberá destinarle más recursos para llevar a cabo controles de *stock* más exhaustivos y complejos, además, realizados de forma periódica y frecuente (Método ABC de inventarios en almacén: que es|AR-Racking, s.f.). Cualquier problema en el inventario de los productos de la Categoría A, como escasez o ruptura de *stock*, supondrá importantes pérdidas a la empresa.

Categoría B: las referencias de producto categorizadas como B en la clasificación ABC son las que tienen una importancia y rotación moderada para la empresa. Generalmente, suponen en torno al 30% del total de productos del almacén y, por norma, no suelen generar más del 20% de los ingresos de la empresa.

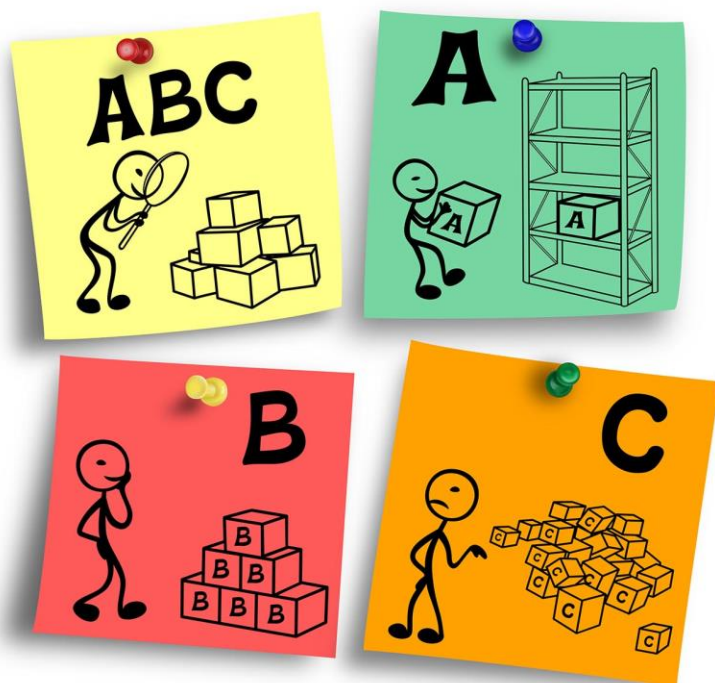
Al ser una categoría intermedia entre la A y la C, se debe revisar periódicamente su estatus, valorando la posibilidad de que se convierta en una referencia de categoría A o C en el futuro.

El control de *stock* de esta categoría de productos también debe realizarse periódicamente, pero en menor frecuencia que los de categoría A que absorberán el grueso de la carga de trabajo en almacén (Método ABC de inventarios en almacén: que es|AR-Racking, s.f.).

Categoría C: según el principio de Pareto que sigue el modelo ABC, las referencias de la categoría C serán las más numerosas, pero también las que menos ingresos aportan a la empresa. Pueden suponer más del 50% de las referencias de productos, pero, en términos de ingresos, no alcanzar ni el 5% del total. Su rotación en el almacén será muy baja, al ser referencias menos demandadas y, por lo tanto, son productos en los que se debe intentar reducir al máximo los recursos destinados a ellos.

El control de inventarios puede ser esporádico y con métodos simples, lo suficiente para evitar problemas de obsolescencia o caducidad, y su ubicación en el almacén será la de los puntos alejados de la zona de expedición, en los niveles superiores o con peor accesibilidad (Método ABC de inventarios en almacén: que es|AR-Racking, s.f.).

Figura 15 : Método ABC

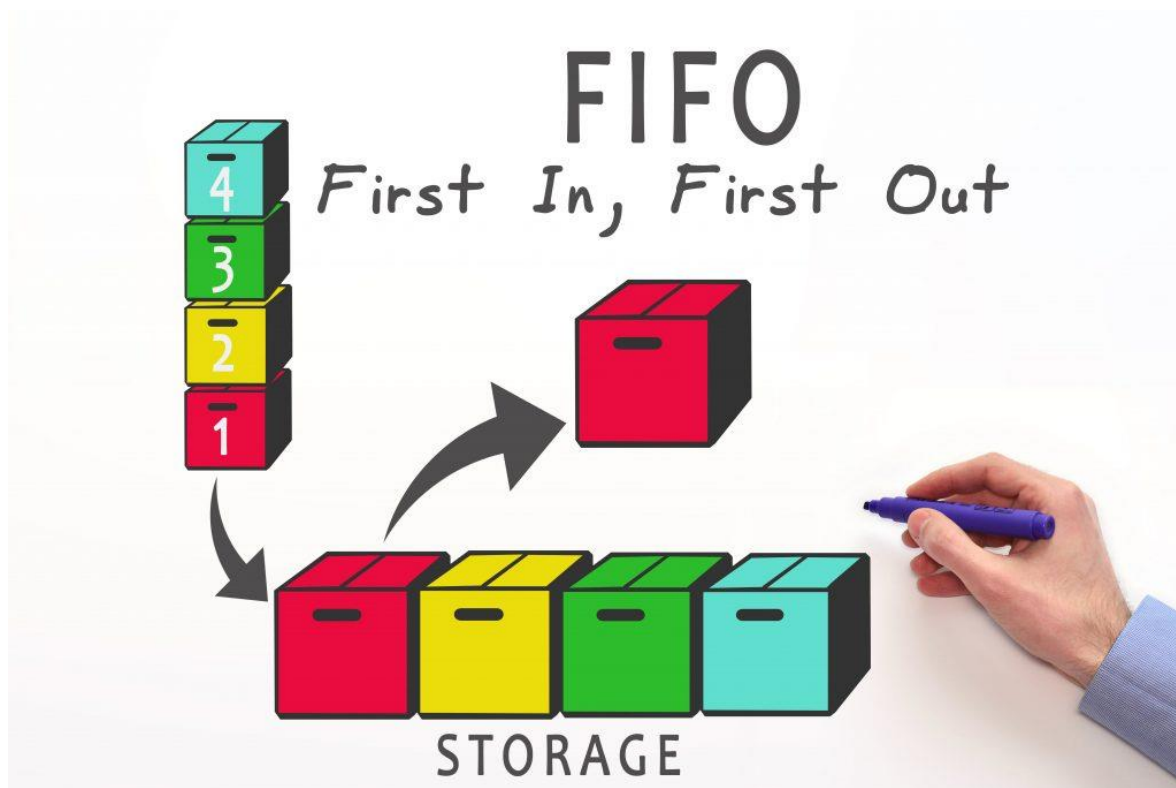


Fuente: (Método ABC | Control de inventarios, s.f.).

Método PEPS (FIFO) aplicado a activos digitales

En ciencias de la computación y teoría de colas, una cola es una estructura de datos que sigue el principio de «primero en entrar, primero en salir» (FIFO, por sus siglas en inglés: First-In, First-Out). Es similar a una línea de espera en la vida cotidiana, donde las personas se colocan en el extremo trasero y se atienden en orden de llegada. Las colas se utilizan en diversos contextos informáticos para gestionar y organizar elementos o tareas en secuencia (Diccionario de Marketing e Internet en general, s.f.).

Figura 16 : Método PEPS (FIFO)



Fuente: (Diccionario de Marketing e Internet en general, s.f.).

Método EOQ adaptado

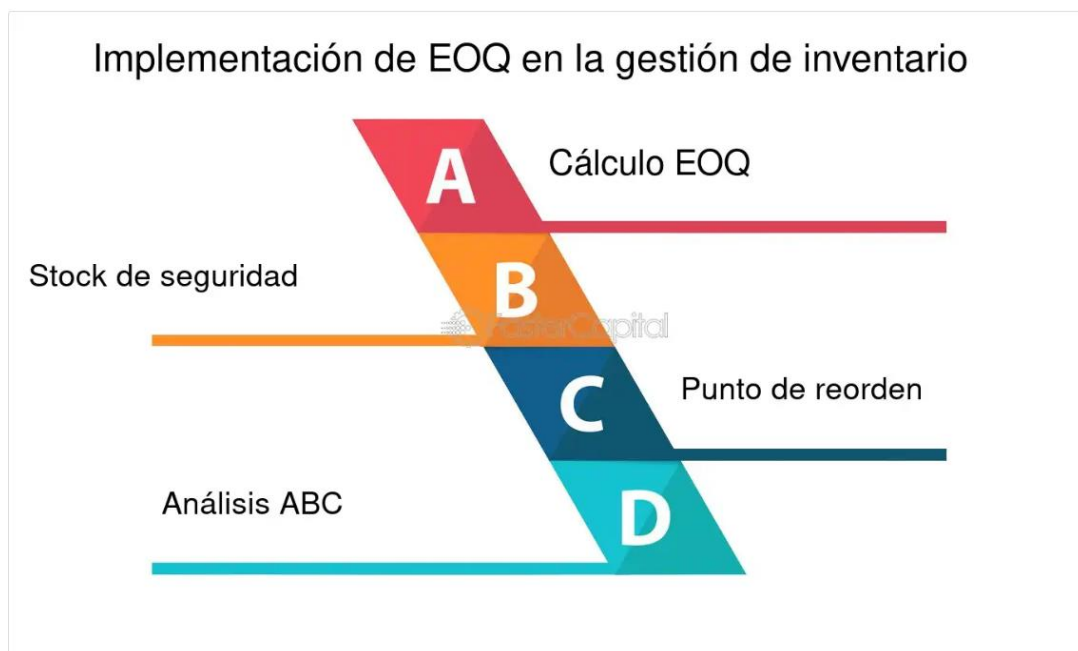
Este modelo se popularizó en 1934 con la publicación de un artículo de la mano de R.H. Wilson, quien da nombre al modelo; pero fue desarrollado en un origen por el ingeniero Ford Whitman Harris cuando trabajaba en la empresa Westinghouse Corporación.

El método surge con el claro objetivo de sistematizar la mercancía que periódicamente se mantiene en el almacén, así como definir la cantidad y la fecha en la que se deben realizar los pedidos a los proveedores. Aunque este sistema se utiliza comúnmente para sistematizar la compra de materia prima, es aplicable a la optimización de compra de cualquier producto

necesario por la empresa, siempre que se pueda determinar los costes de compra en términos de pedido y de almacenamiento

El método es simple y se basa en una fórmula que permite determinar en qué momento y de qué cantidad se tienen que realizar los pedidos de la empresa, teniendo en cuenta la demanda y el *stock* de seguridad mínimo de la compañía (Sistema EOQ o Modelo de Wilson en almacén|Ar-racking.com, 2021).

Figura 17 : Método EOQ adaptado

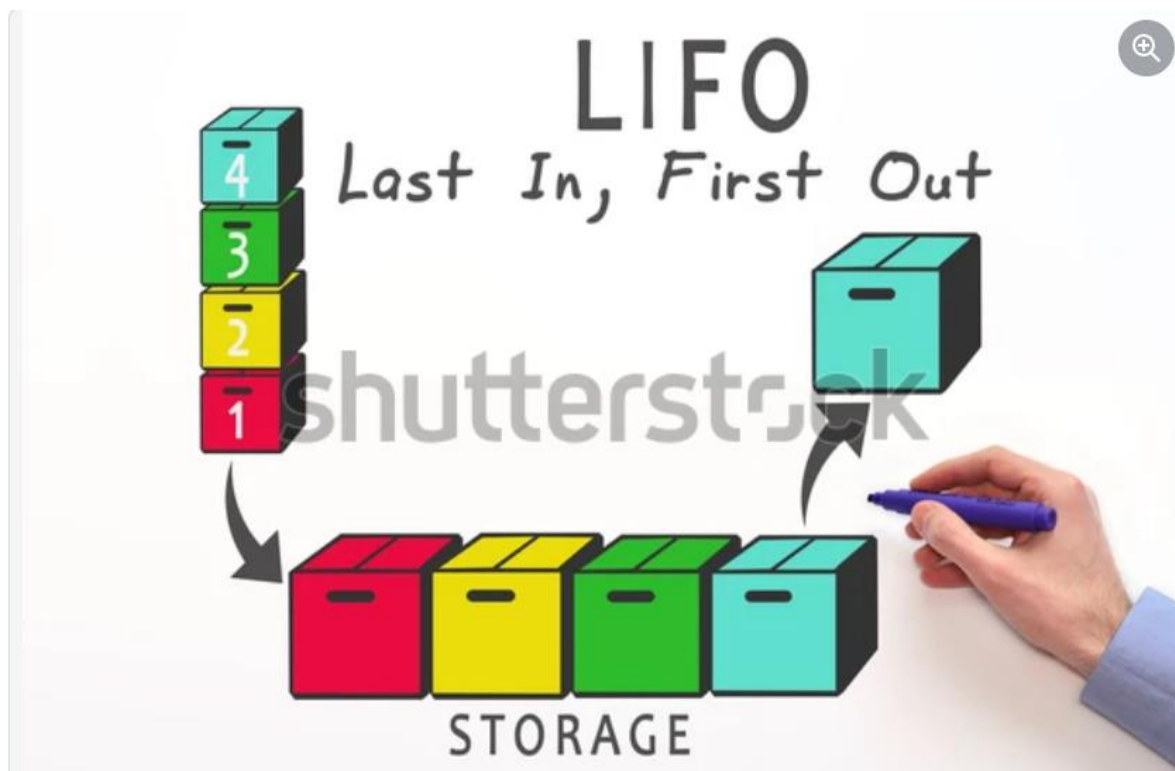


Fuente: (Implementacion de EOQ en la gestión de inventario, s.f.).

Método UEPS (LIFO)

Resulta útil en escenarios donde los activos más recientes deben consumirse o actualizarse primero, por ejemplo, licencias recién adquiridas con vigencias específicas.

Figura 18 : Conteo LIFO



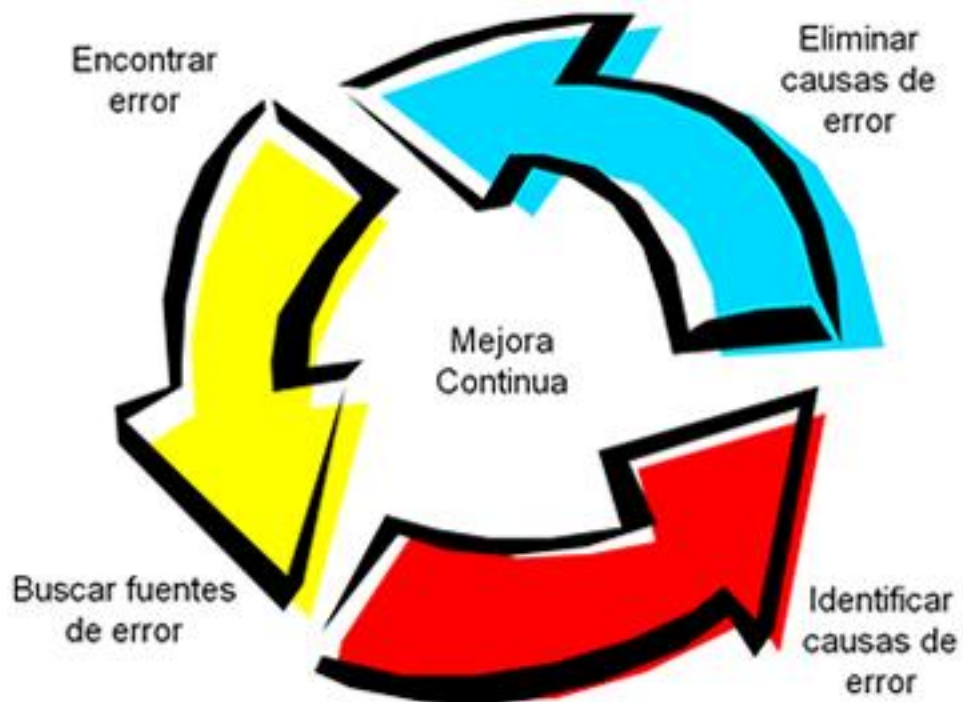
Fuente: (FIFO y LIFO: técnicas de gestión de la carga, 2017).

Conteo cíclico (inventario digital continuo)

Es otro tipo de método para el conteo de inventarios, consiste en el recuento frecuente de una parte del inventario total, con la finalidad de que todo se haya contado, al menos, una vez en un periodo de tiempo determinado. Así mismo, es el complemento de método ABC, ya que a cada clase se le asigna una frecuencia de recuento diferente.

Entre sus beneficios, se encuentran la mejora de la exactitud y fiabilidad del control de los inventarios, permitiendo encontrar y corregir a tiempo ciertas discrepancias que puedan afectar a la empresa, sin requerir de un conteo total de los artículos. Hay que tener en cuenta que el método EOQ y el conteo cíclico o conteo de pedido de producción son modelos que se pueden aplicar solo cuando la demanda es conocida.

Figura 19 : Conteo cíclico

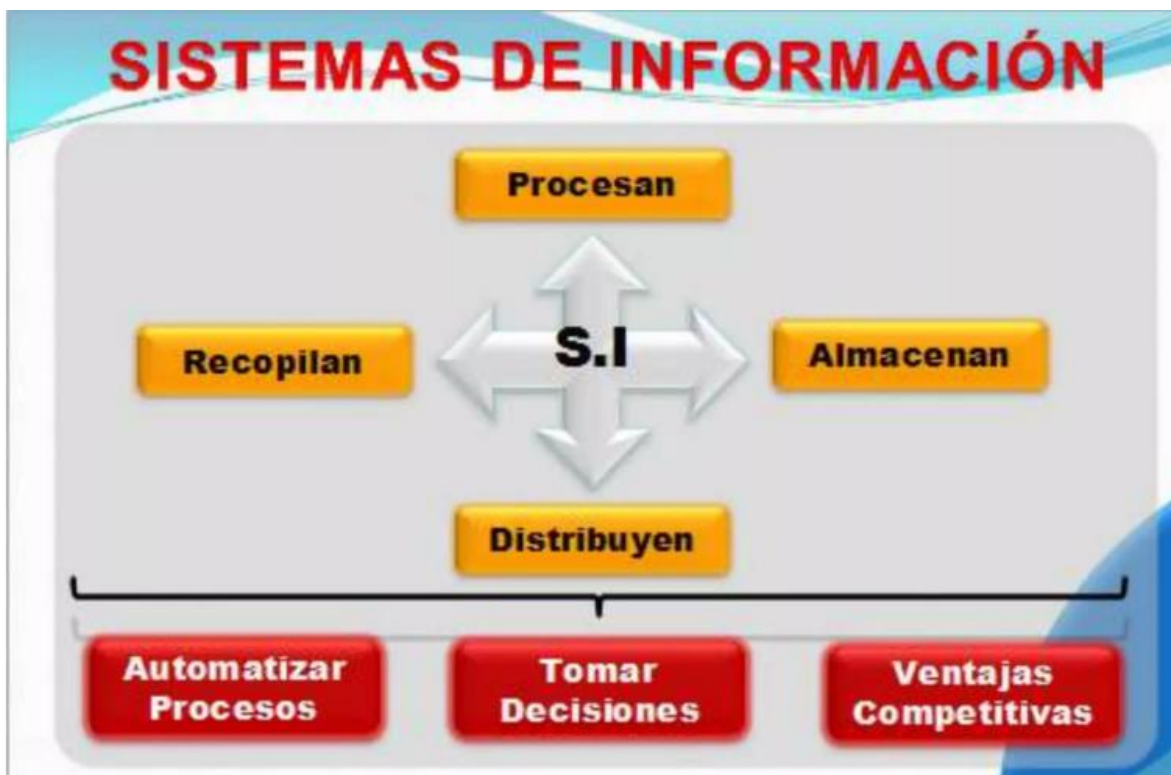


Fuente: (Conteo Cíclico | Bot Verification, s.f.).

Sistemas de información aplicados a la gestión de activos digitales

Son un conjunto de componentes interrelacionados que recopilan datos, los recuperan, procesan y almacenan, siendo su función esencial distribuir información para apoyar la toma de decisiones y control en una organización.

Figura 20: Sistema de Información (SI)



Fuente: (DEFINICIÓN DE SISTEMA DE INFORMACIÓN|sicunisucr, 2014).

Sistemas de procesamiento de transacciones (TPS)

Los sistemas de procesamiento de transacciones en línea (OLTP) se pueden clasificar en dos metodologías principales de procesamiento de información. La elección de TPS de una empresa dependerá de sus necesidades comerciales únicas, mientras que también se puede emplear un modelo híbrido (¿Qué es un sistema de procesamiento de transacciones(TPS)?|IBM, 2024).

Figura 21 : Sistemas de procesamiento de transacciones (TPS)



Fuente: (Sistemas de Procesamiento de Transacciones|Tecnologías-información, s.f.).

Sistemas de información gerencial (MIS)

Un sistema de información de gestión es un conjunto de sistemas y procedimientos que recopilan información de una variedad de fuentes, la compilan y la presentan en un formato legible.

Los administradores utilizan un sistema de información gerencial para crear informes que les brindan una visión general exhaustiva de toda la información que necesitan para tomar decisiones, las cuales van desde minucias diarias hasta estrategias de nivel superior.

Los sistemas de información de gestión de hoy en día dependen en gran medida de la tecnología para compilar y presentar datos, pero el concepto es más antiguo que las tecnologías de cómputo modernas (Sistemas de Información Gerencial - Objetivos, s.f.).

Figura 22: Sistemas de información gerencial (MIS)



Fuente: <https://www.tecnologias-informacion.com/sigerencial.html>

Sistemas de control de procesos de negocio (BPM)

Un sistema de control de procesos de negocio (o *business process management*) es un modelo de gestión organizacional, que usa una serie de herramientas e instrumentos, los cuales analizan los procesos de negocio para optimizarlos.

En este sentido, el BPM orienta sus esfuerzos a estandarizar y automatizar la mayor cantidad posible de tareas, con la finalidad de conseguir un mejor aprovechamiento de los recursos. Para lograrlo, este modelo se vale de programas tecnológicos diseñados para tal fin.

De esta manera, a través de estas tecnologías, se puede monitorear constantemente la evolución y resultado de los procesos de negocio ya implementados, para detectar cualquier oportunidad de mejora y así poder aprovecharla (Sistema de control de procesos de negocio (BPM): Ejemplos y características|Sierra y Yhorman, 2022).

Figura 23 : Sistemas de control de procesos de negocio (BPM)



Fuente: (Sistema de control de procesos de negocio (BPM): Ejemplos y características|Sierra y Yhorman, 2022).

Sistemas de información de marketing (SIM)

En muchas ocasiones, las empresas invierten demasiado esfuerzo y tiempo en la investigación de mercado, pero no pueden procesar de manera adecuada los altos volúmenes de datos a los que se enfrentan. En este punto, deben confiar en un sistema de información de *marketing*, es decir, en una herramienta que permita automatizar esos datos y poder usarlos eficientemente para tomar mejores decisiones (¿Qué es un sistema de información de marketing en la actualidad?|Migallón, 2021).

Figura 24 : Sistemas de información de marketing (SIM)



Fuente: (¿Qué es un sistema de información de marketing en la actualidad?|Migallón, 2021).

Sistemas de colaboración empresarial (ERP)

Son sistemas de planificación empresariales que permiten llevar todos los aspectos básicos de la empresa: contabilidad, facturación, costes, pago de impuestos, nóminas, etc. Es decir, es una herramienta de *software* diseñada para controlar y administrar una empresa. Lo que se pretende es automatizar la mayoría de los procesos (tareas y campos de una base de datos) de manera organizada.

A menudo, se les conoce como *back office*, a la inversa de *front office*, en la medida en que los primeros se ocupan de aspectos administrativos internos, mientras que el segundo término refiere al *software* u operaciones relacionados con atención al cliente y público general (Introducción: ¿Qué es un sistema ERP?|Perfil, s.f.).

Figura 25 : Sistemas de colaboración empresarial (ERP)



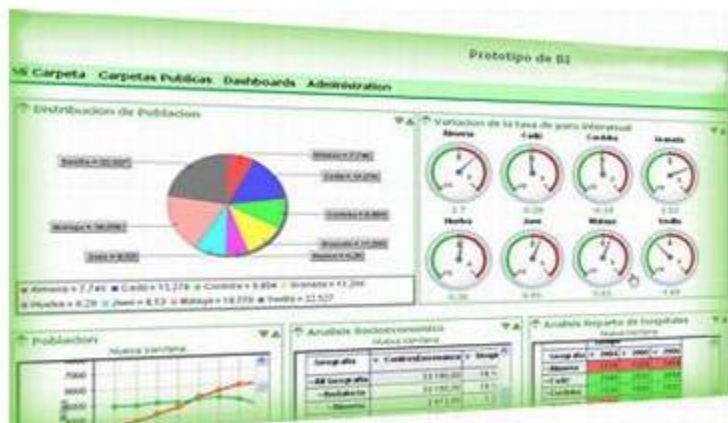
Fuente: (Introducción: ¿Qué es un sistema ERP?|Perfil, s.f.).

Sistemas de soporte a la toma de decisiones (DSS)

Un sistema de soporte a la decisión (DSS) es una herramienta de *Business Intelligence* enfocada al análisis de los datos de una organización. En principio, puede parecer que el análisis de datos es un proceso sencillo y fácil de conseguir mediante una aplicación hecha a medida o un ERP sofisticado. Sin embargo, no es así: estas aplicaciones suelen disponer de una serie de informes predefinidos en los que presentan la información de manera estática, pero no permiten profundizar en los datos, navegar entre ellos, manejarlos desde distintas perspectivas, etc.

El DSS es una de las herramientas más emblemáticas del *Business Intelligence*, ya que, entre otras propiedades, permiten resolver gran parte de las limitaciones de los programas de gestión.

Figura 26 : Sistemas de soporte a la toma de decisiones (DSS)



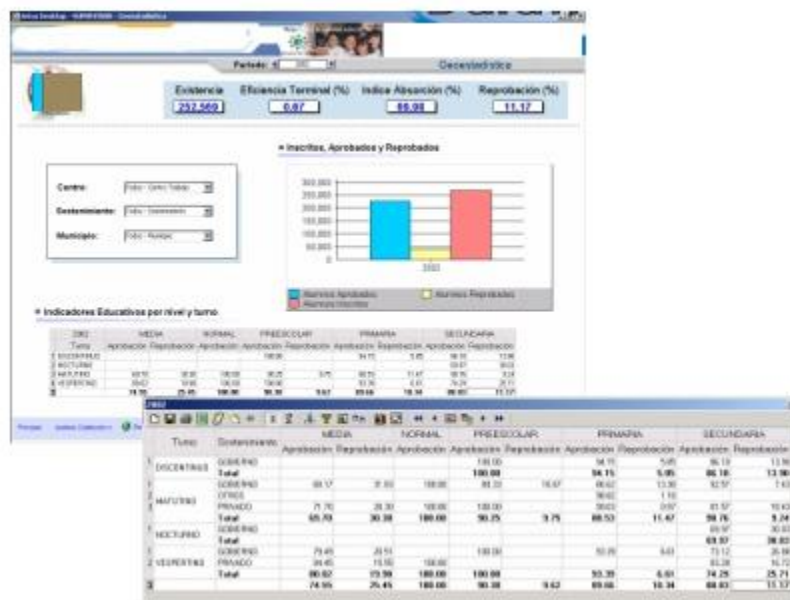
Fuente: (Sistemas de Soporte a la Decisión (DSS)|Sinnexus.com, s.f.).

Sistemas de información ejecutiva (EIS)

Un sistema de información para ejecutivos o sistema de información ejecutiva es una herramienta de *software*, basada en un DSS, que provee a los gerentes de un acceso sencillo a información interna y externa de su compañía, la cual es relevante para sus factores clave de éxito.

La finalidad principal es que el ejecutivo tenga a su disposición un panorama completo del estado de los indicadores de negocio que le afectan al instante, manteniendo también la posibilidad de analizar con detalle aquellos que no estén cumpliendo con las expectativas establecidas, para determinar el plan de acción más adecuado.

De forma más pragmática, se puede definir un EIS como una aplicación informática que muestra informes y listados (*query & reporting*) de las diferentes áreas de negocio, de forma consolidada, para facilitar la monitorización de la empresa o de una unidad de esta (Sistemas de Información Ejecutiva (EIS)|Sinnexus, s.f.).

Figura 27 : *Sistemas de información ejecutiva (EIS)*

Fuente: (Sistemas de Información Ejecutiva (EIS)|Sinnexus, s.f.).

Ciclo de vida de un sistema de información

Es un sistema, automatizado o manual, que engloba a personas, máquinas o métodos organizados para recopilar, procesar o transmitir datos que representan información. Un sistema de información engloba la infraestructura, la organización, el personal y todos los componentes necesarios para la recopilación, procesamiento, almacenamiento, transmisión, visualización, diseminación y organización de esta.

Cualquier sistema de información va pasando por una serie de fases a lo largo de su vida. Su ciclo de vida comprende una serie de etapas, entre las que se encuentran las siguientes:

Planificación: realizar una serie de tareas previas que influirán decisivamente en la finalización con éxito del proyecto.

Análisis de requerimientos: averiguar qué es exactamente lo que tiene que hacer el sistema. La etapa de análisis en el ciclo de vida del *software* corresponde al proceso mediante el cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema.

Diseño: se han de estudiar posibles alternativas de implementación para el sistema de información por construir y se ha de decidir la estructura general que tendrá el sistema (su diseño arquitectónico). El diseño de un sistema es complejo y el proceso debe realizarse de forma iterativa.

Desarrollo: seleccionar las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de sistema que se vaya a construir. La elección de estas herramientas dependerá en gran parte de las decisiones de diseño tomadas hasta el momento y del entorno en el que el sistema deberá funcionar.

Prueba: tiene como objetivo detectar los errores que se hayan podido cometer en las etapas anteriores del proyecto (y, eventualmente, corregirlos). La búsqueda de errores que se realiza en la etapa de pruebas puede adoptar distintas formas, en función del contexto y de la fase del proyecto.

Integración y ejecución: se debe planificar el entorno en el que el sistema debe funcionar, tanto *hardware* como *software*: equipos necesarios y su configuración física, redes de interconexión entre los equipos y de acceso a sistemas externos, sistemas operativos y bibliotecas.

Estas etapas son un reflejo del proceso que se sigue a la hora de resolver cualquier tipo de problema.

Operación y mantenimiento: la etapa de mantenimiento consume típicamente del 40 al 80 % de los recursos de una empresa de desarrollo de *software*. De hecho, con un 60% de media, es probablemente la etapa más importante del ciclo de vida del *software*.

- Eliminar los defectos que se detecten durante su vida útil, lo primero que a uno se le viene a la cabeza cuando piensa en el mantenimiento de cualquier cosa.
- Adaptarlo a nuevas necesidades cuando el sistema ha de funcionar sobre una nueva versión del sistema operativo o en un entorno de *hardware* diferente.
- Añadirle nueva funcionalidad, cuando se proponen características deseables que supondrían una mejora del sistema ya existente. (Ciclo de vida de un sistema de información|Guerrero, 2015)

Figura 28: Ciclo de vida del software



Fuente: (Ciclo de Vida de la Información|Edu.ar, s.f.).

Requerimientos de sistema

En el mundo del desarrollo de software y aplicaciones, los requerimientos son críticos. Son los que definen la funcionalidad y el propósito de una pieza particular de software o aplicación. Sin estos bien definidos, es difícil crear algo que satisfaga las necesidades y expectativas del usuario.

Los requerimientos de *software* son simplemente una descripción de lo que un programa de *software* en particular debe hacer. Actúan como pautas para que los desarrolladores creen un producto funcional que satisfaga las necesidades de los usuarios (Requerimientos en el desarrollo de software y aplicaciones|Northware, 2022).

Requerimientos funcionales

Los requisitos funcionales son especificaciones detalladas de las funciones que un sistema debe realizar para cumplir con las necesidades del usuario. Estos requisitos describen las tareas, procesos y comportamientos que el sistema debe ejecutar.

Por ejemplo, en una aplicación bancaria, un requisito funcional podría ser la capacidad de transferir fondos entre cuentas. Estos requisitos son esenciales para guiar el desarrollo del sistema y asegurar que cumpla con las expectativas del usuario. Además, ayudan a los desarrolladores y diseñadores a entender claramente qué debe hacer el sistema, facilitando la creación de un producto final que satisfaga las necesidades del cliente (Requerimientos funcionales: Ejemplos|Pmoinformatica.com, s.f.).

- **Gestión de usuarios:** crear, editar e inhabilitar perfiles con diferentes roles (administrador, estándar, auditor).

- **Control de inventarios:** registrar y monitorear la entrada y salida de activos o boletas.
- **Auditoría de operaciones:** generar reportes de las bitácoras de inventarios, usuarios, proveedores y conexiones, esto para poder descargar y enviar por correo electrónico para fines de trazabilidad y seguridad.
- **Expedientes de soporte:** dar seguimiento a incidentes, amonestaciones o solicitudes de mantenimiento.
- **Gestión de proveedores:** incluir datos de contacto, historial de compras y evaluación de servicios.
- **Boletas de movimiento:** que se puedan emitir boletas digitales con los movimientos (entrega y retiro de equipos) según corresponda y que las mismas sean enviadas por correo.

Requerimientos no funcionales

Los requerimientos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando.

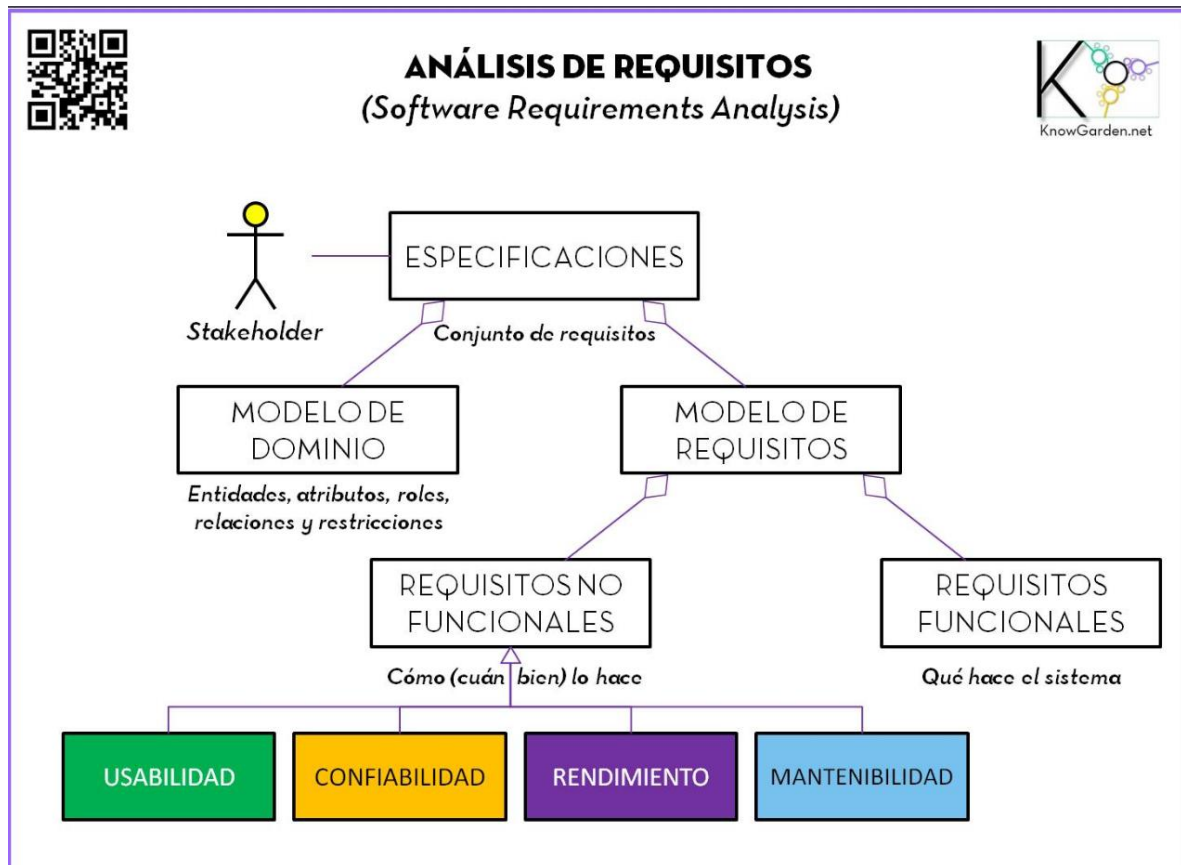
Suelen presentar dificultades en su definición, dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir. Entre los ejemplos de requerimientos no funcionales presentados, se encuentran los referidos a atributos como la eficiencia, seguridad, dependencia y usabilidad del sistema. También están los ejemplos de requerimientos no

funcionales organizacionales y externos (Requerimientos no funcionales:

Ejemplos|Pmoinformatica.com, s.f.).

- El sistema debe ser capaz de procesar N transacciones por segundo. Esto se medirá por medio de la herramienta SoapUI aplicada al *software* Testing de servicios web.
- Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
- El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.
- Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

Figura 29 : Requisitos del sistema



Fuente: (Ingeniería de Requisitos (Software Requirements)|We're always in the KnowGarden, 2015).

Arquitectura

Es la estructura base de un sistema de *software*. Este define cómo se organizan e interactúan sus componentes. Además, es un concepto muy importante para garantizar que el sistema sea escalable, flexible y adaptable a las necesidades de su proyecto. Los tipos de arquitectura más conocidos son:

Monolítica: todos los componentes del sistema están integrados en un solo bloque de código. Este código es simple de implementar, pero puede volverse complejo y difícil de mantener a medida que el sistema crece.

Capas: en esta arquitectura, el sistema se divide en capas lógicas, donde cada capa tiene una responsabilidad específica. Las capas se comunican entre sí a través de interfaces bien definidas. Además, esta arquitectura promueve la modularidad y la separación de preocupaciones (El futuro de la arquitectura de software: tendencias a tener en cuenta|Aicad Business School, 2024).

Microservicios: la arquitectura de microservicios es un enfoque para el desarrollo de *software* que ha ganado popularidad en los últimos años. Se trata de una aproximación a la construcción de aplicaciones que divide el *software* en pequeñas unidades independientes, llamadas microservicios, que se comunican entre sí a través de una red. Esta arquitectura permite a los desarrolladores construir aplicaciones de forma más eficiente, escalable y flexible, y es especialmente útil en el desarrollo de aplicaciones empresariales complejas (Arquitectura de microservicios: una solución para la escalabilidad y flexibilidad de aplicaciones|MyTaskPanel Consulting, 2023).

Eventos: la Programación Orientada a Eventos (POE) es un paradigma de programación que se centra en la interacción entre componentes del sistema a través de eventos. En lugar de

ejecutar un programa de manera lineal, la POE permite que los diferentes componentes del sistema respondan a eventos que ocurren de manera asíncrona, como la interacción del usuario, la llegada de datos externos o cambios en el estado del sistema.

En POE, un evento es una señal que indica que algo ha ocurrido dentro del sistema. Puede ser cualquier cosa, desde una acción del usuario, como hacer clic en un botón, hasta la llegada de datos desde una fuente externa, como una solicitud HTTP. Los eventos pueden desencadenarse de manera síncrona o asíncrona y pueden llevar datos asociados que describen el evento en detalle (Programación Orientada a Eventos: Introducción y conceptos|Canelo, 2024).

Cliente-Servidor: es uno de los estilos arquitectónicos distribuidos más conocidos, el cual está compuesto por dos componentes: el proveedor y el consumidor. El proveedor es un servidor que brinda una serie de servicios o recursos, los cuales son consumidos por el cliente.

En una arquitectura Cliente-Servidor, existe un servidor y múltiples clientes que se conectan al servidor para recuperar todos los recursos necesarios para funcionar. En este sentido, el cliente solo es una capa para representar los datos y se detonan acciones para modificar el estado del servidor, mientras que el servidor es el que hace todo el trabajo pesado.

Componentes: esta armoniosa estructura, que favorece la utilización de herramientas modulares, posibilita la creación de sistemas complejos mediante el uso de bloques ya existentes (Arquitectura Cliente-Servidor|Programación reactiva, s.f.).

Orientada a Servicios (SOA): la arquitectura orientada a servicios (SOA, por sus siglas en inglés) es un método de desarrollo de *software* que utiliza componentes de *software* llamados servicios para crear aplicaciones empresariales. Cada uno de estos servicios brinda una capacidad empresarial, además, pueden comunicarse también con el resto de los servicios mediante diferentes plataformas y lenguajes. Los desarrolladores usan SOA para reutilizar

servicios en diferentes sistemas o combinar varios servicios independientes para realizar tareas complejas (¿Qué es la arquitectura orientada a servicios?|Amazon, s.f.).

Estructura de datos

Una arquitectura de datos describe cómo se gestionan estos, desde la recopilación hasta la transformación, la distribución y el consumo. Establece el plan para los datos y la forma en que fluyen a través de los sistemas de almacenamiento de estos. Además, es fundamental para las operaciones de procesamiento de datos y las aplicaciones de inteligencia artificial (IA).

El diseño de una arquitectura de datos debe estar basado en los requisitos empresariales, que los Data Architect e ingenieros de datos utilizan para definir el modelo de datos respectivo y las estructuras de datos subyacentes, que lo sustentan. Estos diseños suelen facilitar una necesidad empresarial, como una iniciativa de elaboración de informes o de ciencia de datos (¿Qué es la arquitectura de datos?|IBM, 2024).

Existen varios tipos de arquitecturas de datos que se utilizan para gestionar y organizar la información dentro de una organización, cada uno con características y aplicaciones específicas:

Arquitectura de datos centralizada: en este modelo, todos los datos se almacenan en un único repositorio centralizado. Aunque ofrece simplicidad y coherencia, puede ser vulnerable a fallos únicos y representar un cuello de botella en el acceso a los datos (Arquitectura de datos|Datdata.com, 2024).

Arquitectura de datos descentralizada: una base de datos descentralizada es un concepto innovador que utiliza redes *peer-to-peer* o tecnología *blockchain* para almacenar y gestionar datos de manera distribuida. A diferencia de las bases de datos tradicionales, donde los datos son gestionados por una única autoridad o servidor, las bases de datos descentralizadas distribuyen los datos en múltiples nodos, garantizando transparencia, seguridad y resiliencia. (Base de Datos Descentralizada|Coinex.com, s.f.)

Arquitectura de datos federada: para usuarios finales y aplicaciones cliente, los orígenes de datos aparecen como una única base de datos colectiva en el sistema de base de datos. Los usuarios y las aplicaciones interactúan con la base de datos federada gestionada por el servidor federado.

La base de datos federada contiene un catálogo del sistema que almacena información sobre los datos. El catálogo del sistema de la base de datos federada contiene entradas que identifican las fuentes de datos y sus características. Por su parte, el servidor federado consulta la información que está almacenada en el catálogo del sistema de la base de datos federada y el derivador de la fuente de datos, a fin de determinar cuál es el mejor plan para procesar las sentencias de SQL.

El sistema federado procesa sentencias de SQL como si los datos de las fuentes de datos fuesen tablas relacionales ordinarias o vistas dentro de la base de datos federada. Como resultado de ello:

- El sistema federado puede correlacionar datos relacionales con datos en formatos no relacionales. Esto también se aplica cuando las fuentes de datos utilizan distintos dialectos de SQL o cuando no dan soporte a SQL.

- Las características de la base de datos federada tienen prioridad cuando existen diferencias entre las características de la base de datos federada y las de las fuentes de datos. (Base de datos federada|Db2 11.1, 2023)

Arquitectura de datos orientada a servicios (SOA): define una manera de hacer que los componentes de *software* sean reutilizables a través de interfaces de servicio. Los servicios utilizan estándares de interfaz comunes y un patrón arquitectónico para que puedan incorporarse rápidamente a aplicaciones nuevas. De este modo, se evitan determinadas tareas al desarrollador de aplicaciones, que antes debía volver a desarrollar o duplicar la funcionalidad existente o tenía que saber cómo conectar o proporcionar interoperatividad con las funciones existentes.

Cada servicio de una SOA incorpora el código y las integraciones de datos necesarias para ejecutar una función de negocios completa y discreta (por ejemplo, comprobar el crédito de un cliente, calcular un pago de préstamo mensual o procesar una solicitud de hipoteca). Las interfaces de servicio proporcionan acoplamiento suelto, lo que significa que se pueden invocar con poco o ningún conocimiento de cómo el servicio se implementa de manera subyacente, reduciendo las dependencias entre aplicaciones.

Esta interfaz es un contrato de servicios entre el proveedor y el consumidor de servicios. Las aplicaciones detrás de la interfaz de servicios se pueden escribir en Java, Microsoft .Net, Cobol o en cualquier otro lenguaje de programación, suministrado como aplicaciones de *software* empaquetadas por un proveedor (por ejemplo, SAP), aplicaciones SaaS (por ejemplo, Salesforce CRM) u obtenidas como aplicaciones de código abierto. Las interfaces de servicio se definen con frecuencia utilizando un Lenguaje de Descripción de Servicios Web (WSDL), que es una estructura de etiqueta estándar basada en XML (lenguaje de marcado extensible) (¿Qué es la SOA (arquitectura orientada a servicios)?|Ibm.com, 2023).

Arquitectura de datos en la nube: una base de datos en la nube es un servicio de base de datos diseñado y al que se accede a través de una plataforma de *cloud computing*. Cumple muchas de las mismas funciones que una base de datos tradicional con la flexibilidad añadida del *cloud computing*. Los usuarios instalan *software* en una infraestructura de nube para implementar la base de datos.

La gestión de datos de implicación y aplicaciones para redes masivas de usuarios móviles o dispositivos remotos puede suponer todo un reto de escalabilidad y disponibilidad. El problema es que la mayoría de las bases de datos requieren actualizaciones en una base de datos central "maestra". Esto puede provocar cuellos de botella de rendimiento e impedir que las aplicaciones se ejecuten si la conexión a la base de datos maestra no está disponible.

Una base de datos en la nube permite a las organizaciones impulsar el acceso a la base de datos al extremo más alejado de la red para dispositivos móviles, instalaciones remotas, sensores y productos habilitados para internet. Esto ayuda a mejorar la escalabilidad y permite que las aplicaciones sigan funcionando sin conexión (¿Qué es una base de datos en la nube?|Ibm.com, 2023).

Interfaz

Una interfaz de programación también se denomina interfaz de la aplicación o de *software* (términos sinónimos). Permite la conexión de una aplicación o *software* con otro. Una interfaz de programación permite la transmisión de datos y funciones entre sistemas de aplicación y *software*. Actúa como un puente que permite a las aplicaciones comunicarse entre sí mediante el envío de peticiones y la recepción de respuestas, al tiempo que abstrae la complejidad entre bastidores.

Además, las interfaces de programación permiten la portabilidad de los datos, la orientación de las campañas de correo electrónico (por ejemplo, un servicio de correo electrónico de soluciones SaaS para el *marketing* relacional), los datos abiertos, la integración de las funciones de una aplicación (por ejemplo, una aplicación web) y la integración de los datos en una campaña de *marketing* (Interfaz de programación|appleute, 2021).

Componentes

Los componentes pueden ser actualizados o reemplazados de forma independiente, mejorando la flexibilidad y adaptabilidad del sistema. La arquitectura de componentes es útil en sistemas complejos con alta cohesión y bajo acoplamiento entre partes (Arquitectura de componentes|Medium, 2024).

Diagrama UML

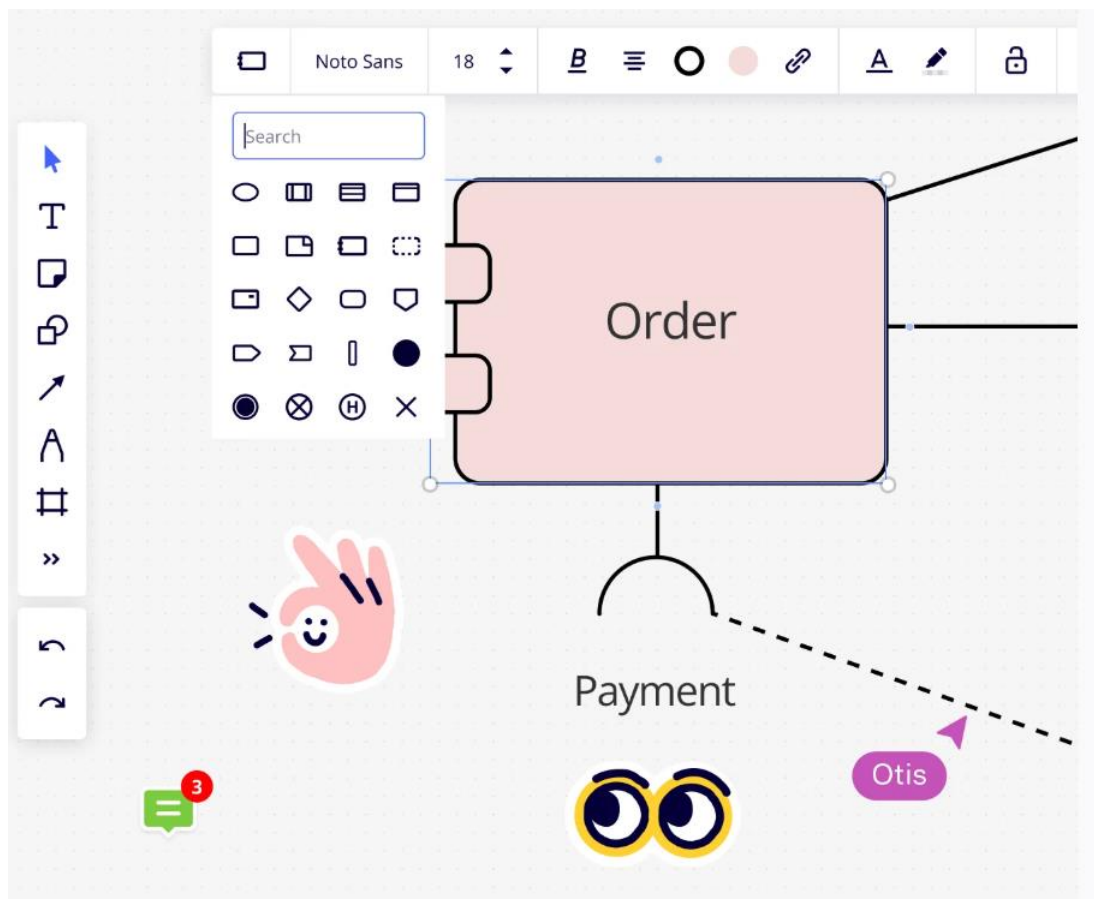
Los diagramas de componentes UML muestran la estructura de los componentes de la arquitectura de un sistema y cómo están conectados e interactúan. Por ejemplo, un sistema de compras en línea tiene pasarelas de pago, catálogos de productos, carritos de compra, etc. Un componente es un elemento de un sistema que puede ser reemplazado sin afectar al resto.

Los diagramas de componentes ofrecen una vista de nivel superior que un diagrama de clases. Mientras que un diagrama de clases muestra clases, sus atributos y métodos, y las relaciones entre objetos, los diagramas de componentes incluyen uno o más de estos y cada componente suele contener múltiples clases.

Quizás lo más importante es que los diagramas de componentes aclaran la estructura de un sistema de *software*. Se pueden ver los componentes del *software* y sus interfaces, así como las dependencias con otros componentes.

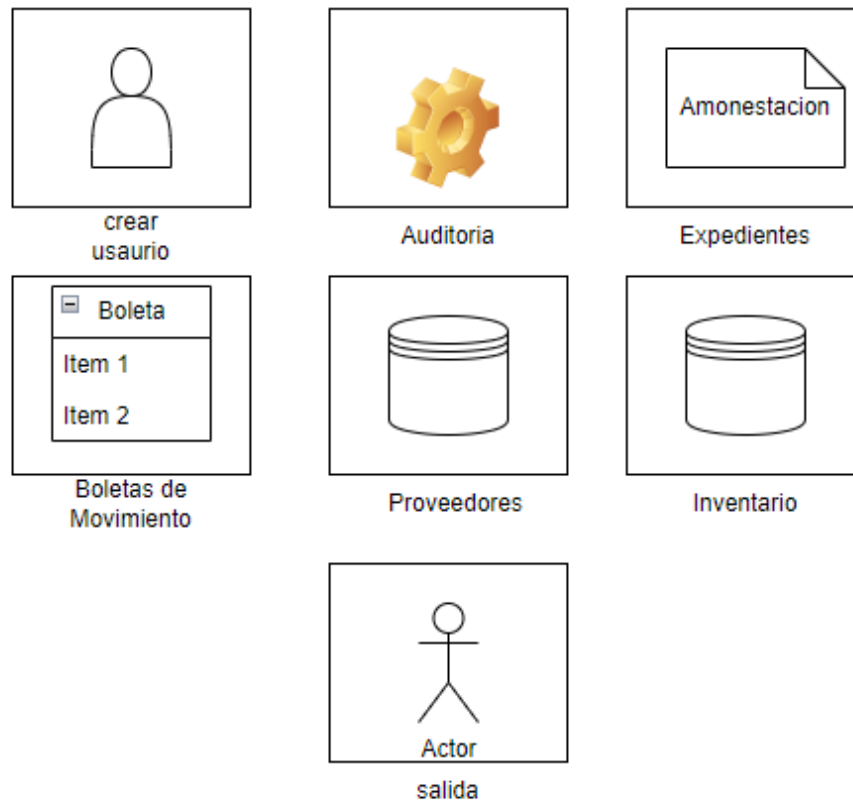
Estos diagramas enseñan la estructura de implementación de un sistema, por lo que los desarrolladores saben cómo mantenerlo y modificarlo. También muestran dependencias entre componentes, lo que puede ayudar a identificar problemas (Diagrama de componentes UML|miro, s.f.).

Figura 30: Componentes



Fuente: (Diagrama de componentes UML|miro, s.f.).

Figura 31: Diagrama UML



Fuente: (Argüello Saborío, B. (2025). Diagrama UML)-

Casos de uso

Un caso de uso es una metodología utilizada en el análisis de sistemas para identificar, aclarar y organizar los requisitos del sistema. El caso de uso está formado por un conjunto de posibles secuencias de interacciones entre sistemas y usuarios en un entorno particular y relacionadas con un objetivo específico. El método crea un documento que describe todos los pasos que sigue un usuario para completar una actividad.

Los analistas de negocios suelen ser responsables de redactar casos de uso y se les emplea durante varias etapas del desarrollo de *software*, como la planificación de los requisitos del sistema, la validación del diseño, la prueba del *software* y la creación de un esquema para la ayuda en línea y los manuales de usuario. Un documento de caso de uso puede ayudar al equipo de desarrollo a identificar y comprender dónde pueden ocurrir errores durante una transacción para que puedan resolverlos.

Registrar un nuevo activo

- **Actor:** Administrador
- **Flujo:** El administrador ingresa los datos del activo, selecciona la categoría, guarda la información.

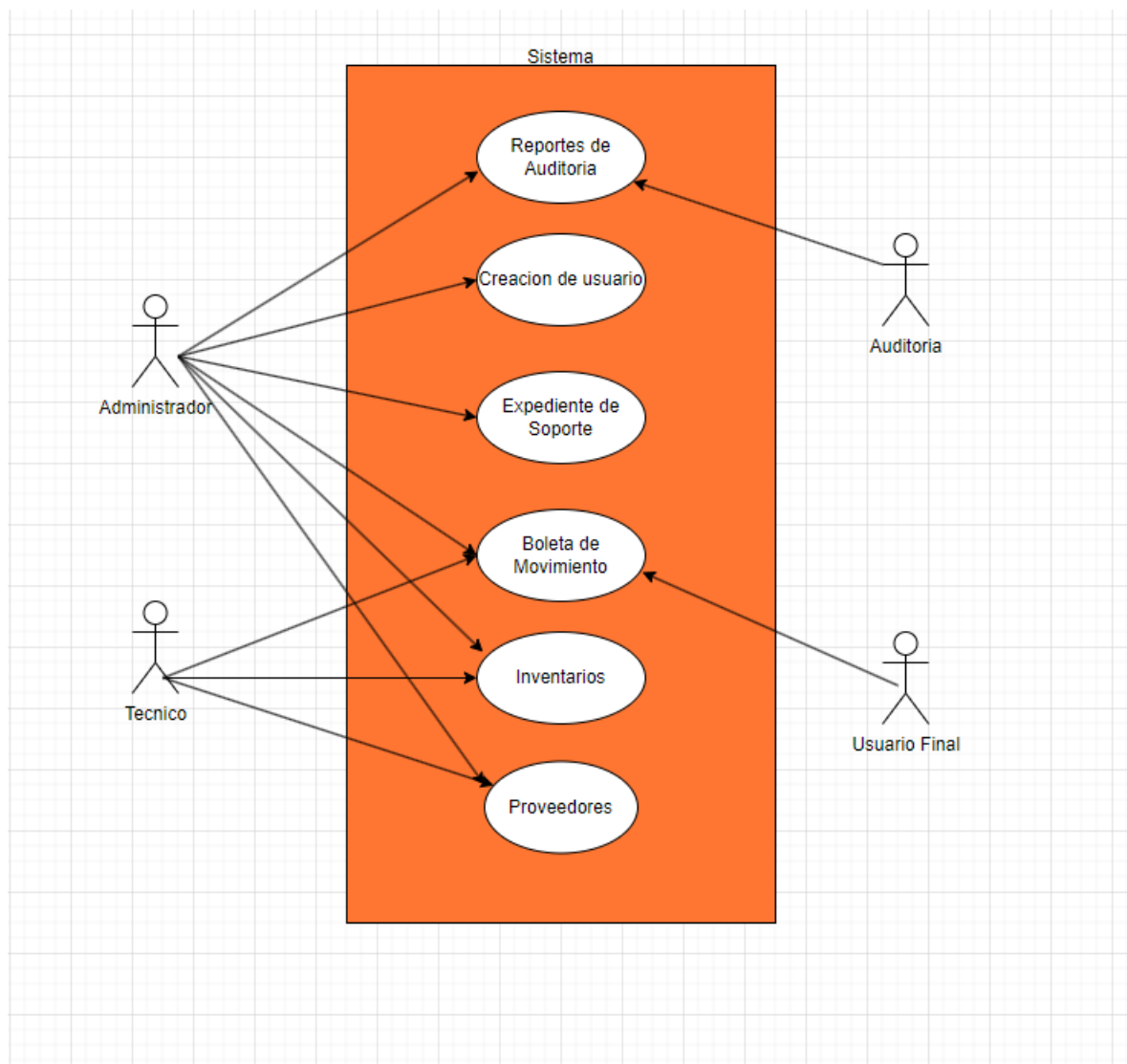
Generar reporte de auditoría

- **Actor:** Gerente / Auditor
- **Flujo:** Solicita un informe detallado de movimientos en un rango de fechas, el sistema procesa y genera un PDF.

Crear expediente de soporte

- **Actor:** Técnico
- **Flujo:** El técnico registra el incidente, adjunta evidencias y actualiza el estado del caso.

Figura 32 : Diagrama caso de uso



Fuente: (Argüello Saborío, B. (2025). Diagrama Caso de Uso).

Base de datos en un gestor de activos digitales

¿Qué es Microsoft SQL Server?

Microsoft SQL Server es uno de los principales sistemas de gestión de bases de datos relacional del mercado que presta servicio a un amplio abanico de aplicaciones de *software* destinadas a la inteligencia empresarial y análisis sobre entornos corporativos.

Basada en el lenguaje Transact-SQL, incorpora un conjunto de extensiones de programación propias de lenguaje estándar y su aplicación está disponible para usarse tanto a nivel con *premise* o bajo una modalidad *cloud* (¿Qué es Microsoft SQL Server y para qué sirve?|Pérez, 2021).

Para el gestor de activos digitales, se ha optado por Microsoft SQL Server en su versión local, dada su estabilidad, seguridad y excelente integración con el ecosistema .NET. Esta elección permite administrar el inventario de activos (digitales o físicos asociados), mantener un registro detallado de usuarios y roles, así como llevar la trazabilidad de operaciones y auditorías.

Figura 33 : SQL Server

¿Qué es SQL?

SQL es el acrónimo de *Structured Query Language* o Lenguaje de Consulta Estructurado

El **SQL** es el lenguaje estándar *ANSI/ISO* de **definición, manipulación y control** de bases de datos relacionales, y es soportado prácticamente por todos los manejadores relacionales disponibles en el mercado.

Es un lenguaje **declarativo**, muy parecido al lenguaje natural (*Inglés*) y es muy expresivo. Por estas razones y como estándar, el SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales.

Permite al usuario **crear, leer, actualizar y eliminar** tablas y bases de datos relacionales.

Además, permite realizar **consultas** de distintas formas, utilizando declaraciones similares al inglés

1.620 x 911

Fuente: (Explorando los lenguajes de interrogación de bases de datos: El estándar ANSI SQL|Cheque Formacion, 2024).

Metodología de desarrollo

Una metodología de desarrollo de *software* es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de desarrollo de sistemas de información. Una gran variedad de estos marcos de trabajo ha evolucionado durante los años, cada uno con sus propias fortalezas y debilidades. Sin embargo, una metodología de desarrollo de sistemas no tiene que ser necesariamente adecuada para usarla en todos los proyectos. Cada una de las metodologías disponibles es más adecuada para tipos específicos de proyectos, basados en consideraciones técnicas, organizacionales, de proyecto y de equipo (Metodologías de desarrollo de software|de Licenciatura en Sistemas y Computación, s.f.).

Metodologías recomendadas

Scrum

- ***Sprints***: facilita la incorporación progresiva de funciones, como el módulo de informes o el módulo de seguridad, lo que simplifica la provisión constante de valor al cliente.
- ***Reuniones de retroalimentación***: promueve la comunicación continua con los interesados y la verificación precoz de los elementos, garantizando que el producto final satisfaga las expectativas.

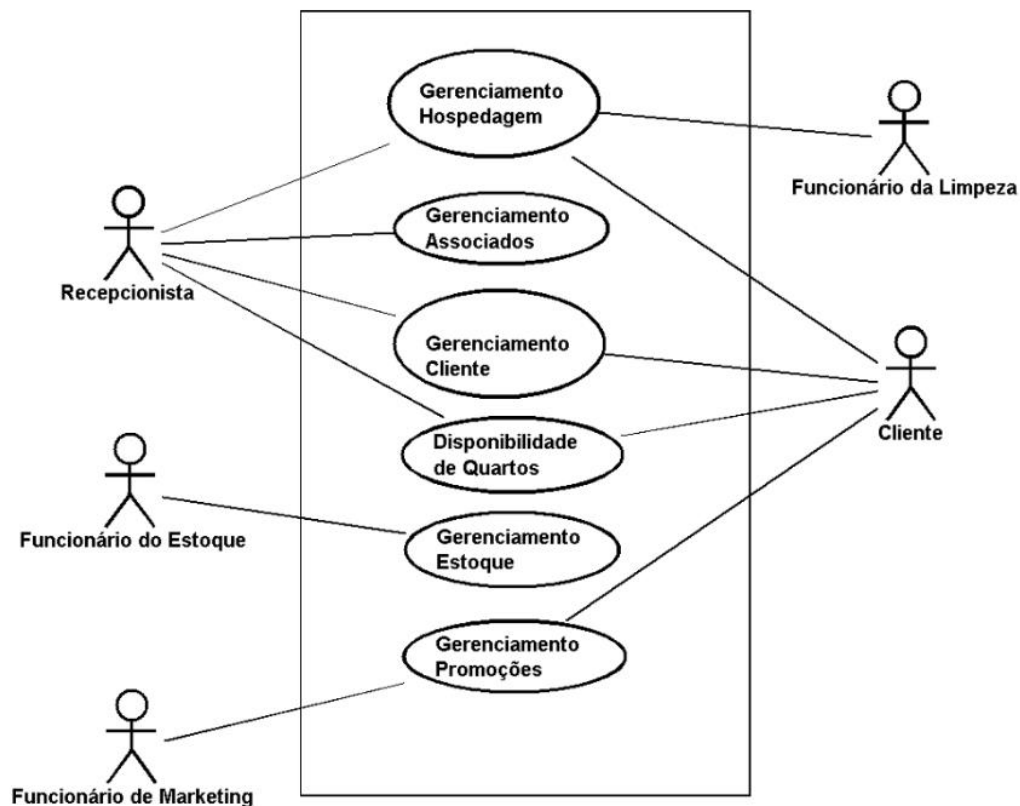
Incremental

- ***Desarrollo por fases:*** se elabora un primer prototipo centrado en la administración de usuarios y roles, para luego desarrollar módulos extra, como el inventario y la auditoría.
- ***Pruebas tempranas:*** cada avance funcional es objeto de evaluaciones unitarias e integradas, lo que simplifica la identificación precoz de fallos y disminuye los riesgos en etapas avanzadas del proyecto.

Casos de uso clave

Los casos de uso se crean para refinar un conjunto de requisitos basados en un rol o tarea. En lugar de la lista tradicional de requisitos que pueden no abordar directamente el uso de la solución, los casos de uso agrupan requisitos comunes basados en el tipo de rol u objetivo. Además, definen lo que los usuarios o roles están haciendo en la solución, un proceso de negocio define cómo realizan estas funciones (Definición de casos de uso|IBM Product Master 12.0.0, 2025).

Figura 34: Ejemplo de un caso de uso



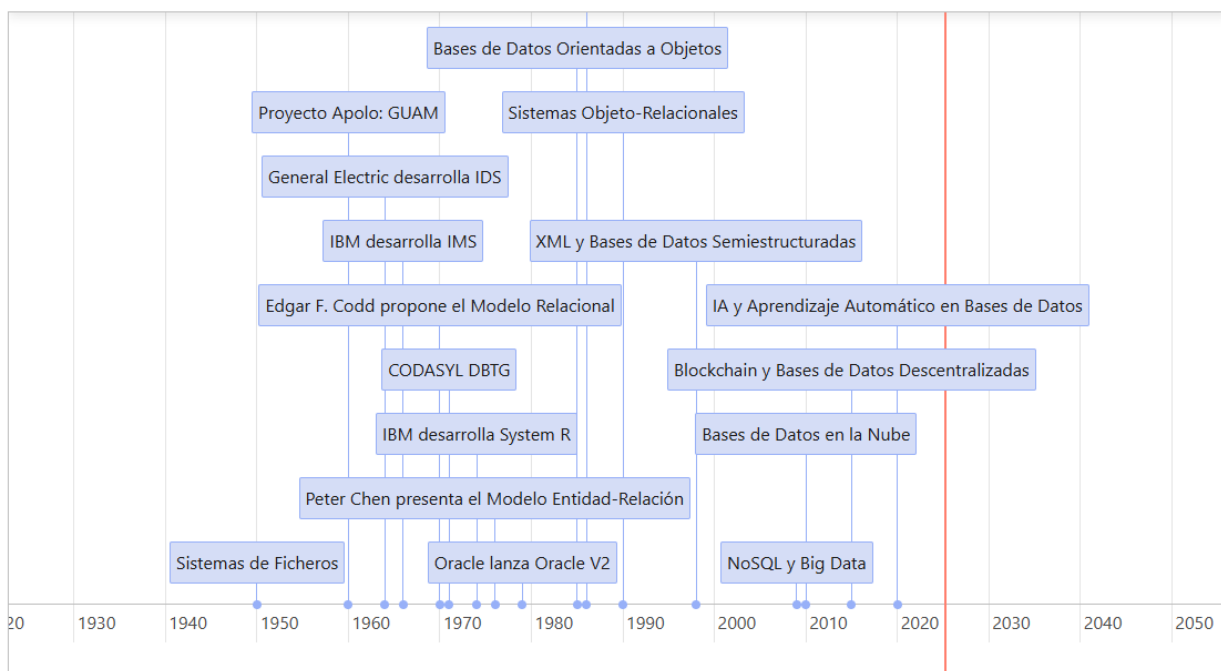
Fuente: (Diagrama de casos de Uso|Fandom Inc., s.f.).

Base de datos

Historia de las bases de datos: un viaje a través del tiempo

En la era digital actual, las bases de datos son el corazón de casi todos los sistemas de información. Desde las aplicaciones móviles usadas diariamente hasta los complejos sistemas empresariales que mueven la economía global, las bases de datos juegan un papel crucial en el almacenamiento, organización y recuperación de información.

Figura 35: Historia de las bases de datos.



Fuente: (Historia de las Bases de Datos: Un Viaje a través del Tiempo[Mosquera, 2024]).

Los precursores: sistemas de ficheros

Antes de que existieran las bases de datos como se conocen hoy, el mundo de la informática dependía de los sistemas de ficheros. Estos sistemas consistían en conjuntos de ficheros de datos y programas de aplicación que permitían a los usuarios trabajar con ellos. Aunque simples en comparación con las bases de datos modernas, los sistemas de ficheros sentaron las bases para lo que vendría después (Historia de las Bases de Datos: Un Viaje a través del Tiempo[Mosquera, 2024]).

Figura 36: Base de datos



Fuente: (¿Qué es la integración de bases de datos? Descripción general y beneficios|Naeem, 2019).

¿Qué es una base de datos de código abierto?

Una base de datos de código abierto es cualquier aplicación de base de datos con un código base que es gratis de ver, descargar, modificar, distribuir y reutilizar. Las licencias de código abierto les otorgan a los desarrolladores la libertad de desarrollar nuevas aplicaciones usando las tecnologías de base de datos existentes.

Ejemplos de bases de datos de código abierto

Ya sea que su objetivo es almacenar datos estructurados de las consultas de SQL o datos sin estructurar a través de objetos JSON, existen varias soluciones de bases de datos para elegir en la web.

Entre los ejemplos de bases de datos relacionales más comúnmente usadas se incluyen los siguientes:

- **MySQL:** es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que emplea el lenguaje SQL para administrar y recuperar datos de manera eficiente. Este sistema es ampliamente adoptado en el desarrollo de aplicaciones web debido a su capacidad para manejar grandes volúmenes de información y su compatibilidad con diversas plataformas (Qué es MySQL: Características y ventajas|Robledano, 2019).
- **PostgreSQL:** PostgreSQL es un viejo conocido en el campo de las bases de datos *open source*. Al fin y al cabo, los orígenes del sistema de gestión de bases de datos relacionales por objetos se remontan a los años 80. Sin embargo, no está en absoluto anticuado. Por el contrario, el *software* bajo licencia BSD ha podido crecer y mejorar constantemente a lo largo de los años y es ampliamente compatible con el estándar SQL (Comparativa de 5 bases de datos open source|IONOS Digital Guide, s.f.).
- **MariaDB:** el servidor de bases de datos MariaDB es obra de los mismos desarrolladores que MySQL y también incluye todos los parches de seguridad de la base de datos más antigua. MariaDB funciona de forma similar en términos de uso y también puede ofrecer funciones de MySQL. Las funciones de pago mencionadas pueden seguir siendo de uso gratuito (Comparativa de 5 bases de datos open source|IONOS Digital Guide, s.f.).

Entre los ejemplos de las bases de datos NoSQL más comúnmente usadas, se incluyen los siguientes:

- **MongoDB:** MongoDB es otra base de datos NoSQL. Destaca en el manejo de aplicaciones móviles, catálogos de productos y gestión de contenidos, entre otras. La base de datos basada en documentos funciona en varios sistemas operativos, por lo que los datos se almacenan con MongoDB en el formato BSON en las llamadas colecciones. Como estas se distribuyen en diferentes servidores, se aumenta la disponibilidad y la carga de datos es soportada por varios servidores (Comparativa de 5 bases de datos open source|IONOS Digital Guide, s.f.).
- **CouchDB:** le permite acceder a sus datos donde los necesite. El protocolo de replicación Couch se implementa en una variedad de proyectos y productos que abarcan todos los entornos informáticos imaginables, desde clústeres de servidores distribuidos globalmente hasta teléfonos móviles y navegadores web (Apache CouchDB|Apache.org, s.f.).
- **Almacene sus datos de forma segura en sus propios servidores o con cualquier proveedor líder de servicios en la nube.** Sus aplicaciones web y nativas adoran CouchDB, porque habla JSON de forma nativa y admite datos binarios para todas sus necesidades de almacenamiento de datos.
- **Cassandra:** sobre todo con volúmenes de datos muy grandes, las bases de datos *open source* pueden alcanzar rápidamente sus límites. Una excepción explícita en este sentido es Apache Cassandra. El sistema de gestión de bases de datos está basado en Java e impresiona por su nula tolerancia a los tiempos de inactividad.

En 2008, el sistema fue lanzado por Facebook y hoy tiene su propio lenguaje de consulta. Apache Cassandra es una base de datos NoSQL orientada a columnas y distribuye las enormes cantidades de datos a diferentes clústeres. Apache Cassandra también tiene una alta puntuación en análisis y registro en comparación con otras *open source databases* (Comparativa de 5 bases de datos open source|IONOS Digital Guide, s.f.).

Cómo elegir la base de datos de código abierto correcta para sus necesidades

Existen varias formas de elegir la base de datos correcta para sus necesidades. Puede analizar la documentación o comunicarse con la empresa, si la documentación aún no está disponible públicamente. También puede usar foros y blogs en línea, así como el código fuente como ayuda para determinar qué base de datos de código abierto puede ser la adecuada (¿Qué es una base de datos de código abierto?|Purestorage, 2021).

Figura 37: Base de datos Open Source



Fuente: (Base de datos Open Source|ITD Consulting, 2022).

Origen de Microsoft SQL Server

El registro de errores de SQL Server es un archivo de texto estándar que contiene información y mensajes de error de Microsoft SQL Server. WinCollect supervisa el registro de errores para sucesos nuevos y reenvía los sucesos al producto QRadar®.

Sumado a lo anterior, el registro de errores proporciona información significativa para ayudarle a resolver problemas o alertarle de problemas potenciales o existentes. La salida del registro de errores incluye la hora y la fecha en que se registró el mensaje, el origen y la descripción de este. Si se produce un error, el registro contiene el número de mensaje de error y una descripción. Además, Microsoft SQL Servers conserva copias de seguridad de los últimos seis archivos de registro de errores.

La base de datos elegida para el desarrollo del proyecto de Gestión de Activos Digitales es SQL server Local, ya que es la más versátil con las necesidades del sistema en cuestión y destaca que es compatible con Visual Studio. Además, es más rápido en cuanto a rendimiento, escalabilidad y es compatible con las tecnologías existentes en la empresa. SQL server local es la base con la que cuenta la organización (IBM Security QRadar Log Insights|Ibm.com, 2024).

Metodologías de desarrollo

Las metodologías de desarrollo de *software* son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de *software* informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos

desarrollen las funciones de un programa de la mejor manera posible (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

Metodologías de desarrollo de software tradicionales

Las metodologías de desarrollo de *software* tradicionales se caracterizan por definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de *software*. Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios, al contrario que las metodologías ágiles; lo que ha propiciado el incremento en el uso de las segundas (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

Cascada

Es una metodología en la que las etapas se organizan de arriba a abajo, de ahí el nombre. Se desarrollan las diferentes funciones en etapas diferenciadas y obedeciendo un riguroso orden. Antes de cada etapa, se debe revisar el producto a fin de ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predispuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

Incremental

En esta metodología de desarrollo de *software*, se va construyendo el producto final de manera progresiva. En cada etapa incremental, se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida en comparación con el modelo en cascada. El *software* se puede empezar a utilizar incluso antes de que se complete totalmente y, en general, es mucho más flexible que las demás metodologías (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

Prototipo

Se basa en la construcción de un prototipo de *software* que se construye rápidamente para que los usuarios puedan probarlo y aportar *feedback*. Así, se puede arreglar lo que está mal e incluir otros requerimientos que puedan surgir. Es un modelo iterativo que se basa en el método de prueba y error para comprender las especificidades del producto (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

Metodologías ágiles

Scrum

Es también una metodología incremental que divide los requisitos y tareas de forma similar a Kanban. Se itera sobre bloques de tiempos cortos y fijos (entre dos y cuatro semanas) para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (*planning sprint*), ejecución (*sprint*), reunión diaria (*daily meeting*) y demostración de resultados (*sprint review*). Cada iteración por estas etapas se denomina también *sprint* (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

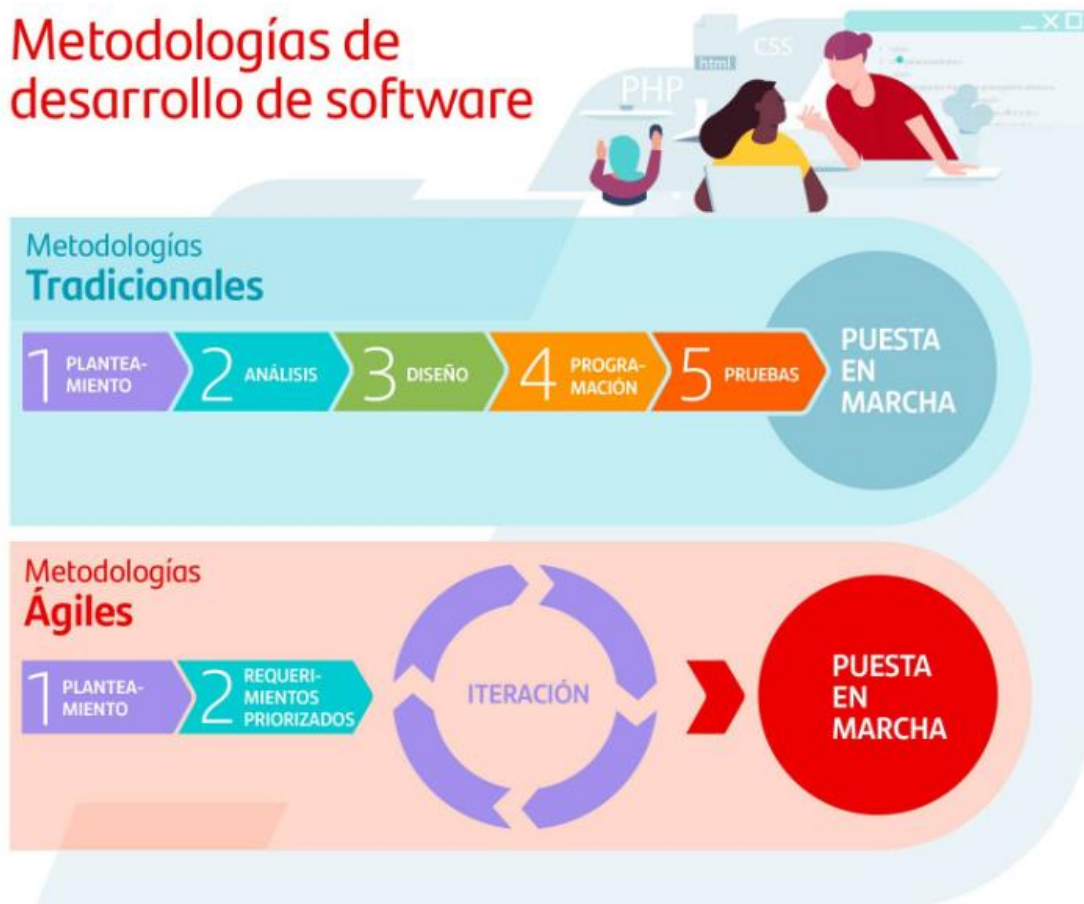
Kanban

Metodología de trabajo inventada por la empresa de automóviles Toyota. Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo dividido en tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual basado en tareas prioritarias e incrementando el valor del producto (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

Programación extrema (Extreme programming, XP)

Es una metodología de desarrollo de *software* basada en las relaciones interpersonales, que se consideran la clave del éxito. Su principal objetivo es crear un buen ambiente de trabajo en equipo y que haya un *feedback* constante del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, *testing*, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente *in situ*, entregas frecuentes y planificación (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

Figura 38: Metodologías ágiles y tradicional



Fuente: (Metodologías de desarrollo de software: ¿qué son?|Santanderopenacademy.com, s.f.).

¿Qué es Windows Form?

- Windows Forms es un subconjunto de la .NET Framework Class Library que permite el desarrollo de aplicaciones de escritorio ricas bajo Microsoft Windows.
- Incluye clases base, interfaces, enumeraciones y controles gráficos diversos.

¿Qué es un formulario?

- Un formulario Windows Forms actúa como interfaz del usuario local de Windows.
- Los formularios pueden ser ventanas estándar, interfaces de múltiples documentos (MDI), cuadros de diálogo, etc.
- Los formularios son clases que exponen propiedades, métodos que definen su comportamiento y eventos que definen la interacción con el usuario.

Cómo crear un proyecto de aplicación para Windows

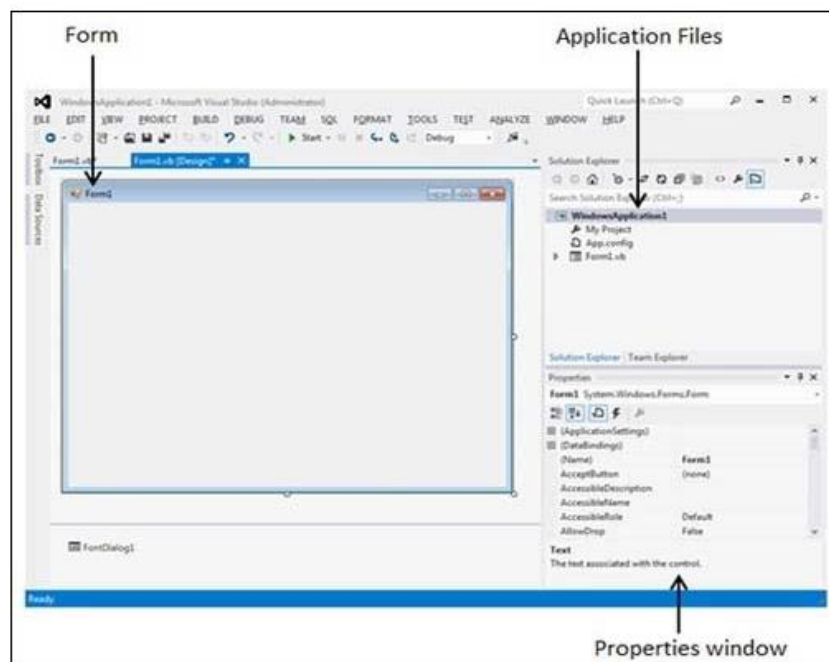
Utiliza los entornos de desarrollo:

- Visual Basic 2022
- Lenguaje Visual Basic Windows Form. Net

- En el menú fila, seleccionar *new project*.
- En la ventana que se despliega, selecciona Windows Application.

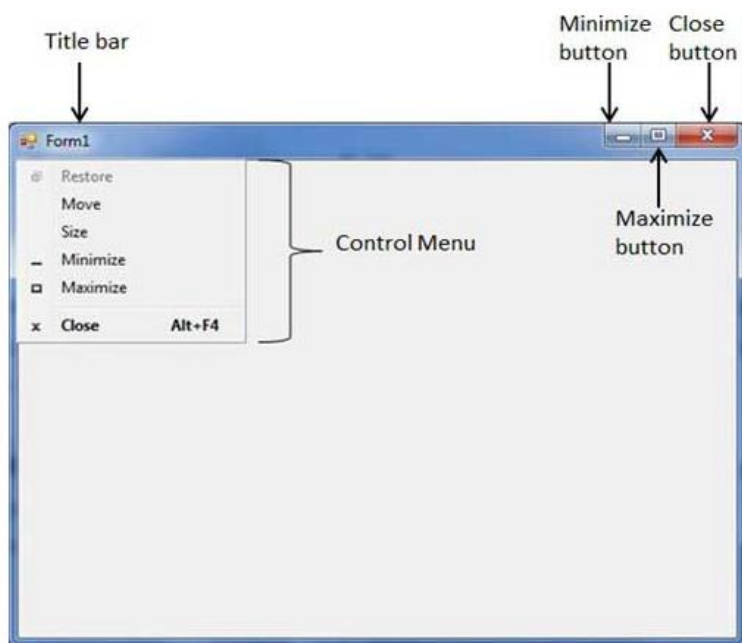
Luego de esto, se abrirá el proyecto nuevo donde se empezará a realizar los diferentes form del proyecto, las clases, módulos y todo lo necesario para el proyecto (Introducción al desarrollo de aplicaciones de escritorio con.net[Eafranco.com, s.f.).

Figura 39: Ejemplo de pantallas de VB.NET-Forms



Fuente: (Ejemplo de pantallas de VB.NET Windows Forms|VB Net Forms, s.f.).

Figura 40: Ejemplo de pantalla VB.NET



Fuente: (Ejemplo de pantallas de VB.NET Windows Forms|VB Net Forms, s.f.).

Cuatro pasos para crear una aplicación básica de Windows Forms

Microsoft Windows Forms ® (WinForms) es una biblioteca que permite desarrollar aplicaciones sofisticadas con interfaz gráfica de usuario (GUI). Esta biblioteca forma parte de .NET Framework. Por lo tanto, una aplicación WinForms de este tipo se puede desarrollar con cualquier lenguaje .NET (C#, F#, VB, etc.) y ejecutarse en cualquier plataforma compatible con .NET Framework. No solo se puede ejecutar en Windows, sino también en Linux o Mac OS X. Por lo tanto, WinForms se considera multiplataforma. Sin embargo, es una tecnología bastante antigua. Para nuevos proyectos, se puede considerar utilizar Windows Presentation Foundation ® (WPF).

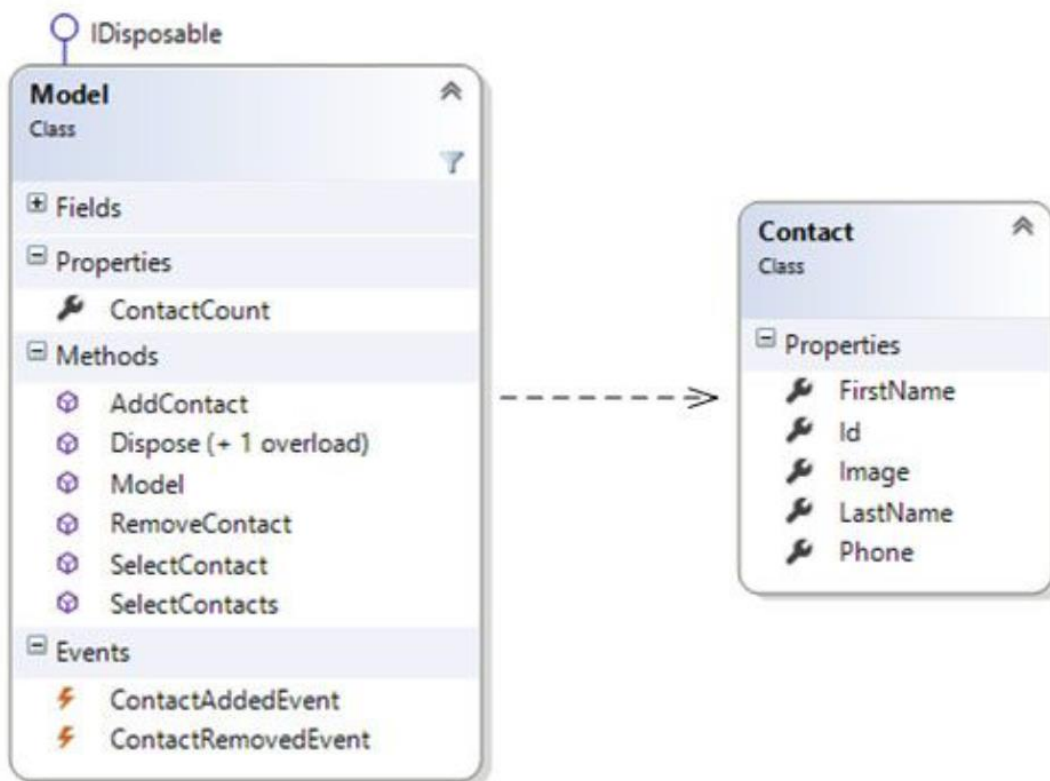
Paso 1. Diseñar la arquitectura de la aplicación

Suponiendo que la especificación de la aplicación está lista antes de diseñar la arquitectura de esta. Para separar la lógica empresarial de su presentador, se sugiere utilizar una

arquitectura similar a MVC. Esto le permitirá modificar cada parte del *software* de forma independiente.

En la aplicación de demostración, un usuario interactúa con el modelo a través de la interfaz gráfica de usuario y el modelo actualiza la interfaz gráfica de usuario a través de los eventos. Esta interacción proporciona un acoplamiento flexible entre los componentes de la aplicación. Por lo tanto, la solución de la aplicación de demostración constará de los dos proyectos siguientes: PhoneBookGUI y PhoneBookModel.

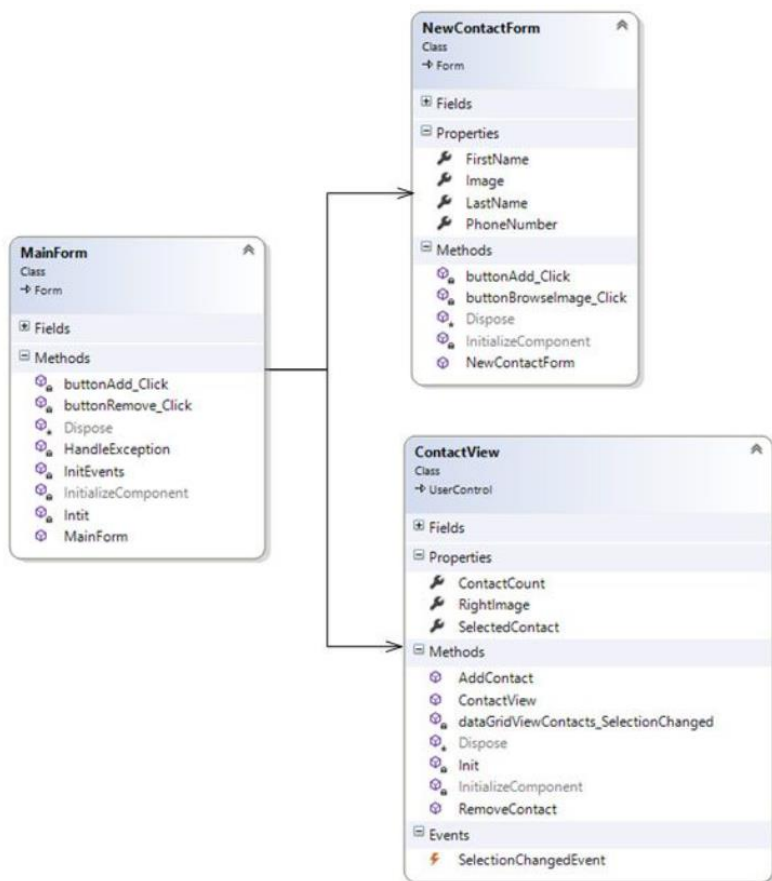
Figura 41: Diagrama de clases del modelo



Fuente: (4 steps to create a basic windows forms application|Apriorit, 2016).

La interfaz gráfica de usuario de la aplicación consta del formulario principal, que contiene referencias a los controles NewContactForm y ContactView. Además, tiene dos botones. El primer botón permite agregar un nuevo contacto con NewContactForm y el segundo botón permite eliminar un contacto seleccionado en el control ContactView. Los formularios heredan la clase System.Windows.Forms y ContactView hereda la clase System.Windows.Forms.UserControl.

Figura 42: Diagrama de clases de la GUI



Fuente: (4 steps to create a basic windows forms application|Apriorit, 2016).

Paso 2. Implementar la lógica del modelo

El modelo debe implementarse de acuerdo con la arquitectura diseñada. Este envuelve el código que trabaja con la base de datos del resto de la aplicación.

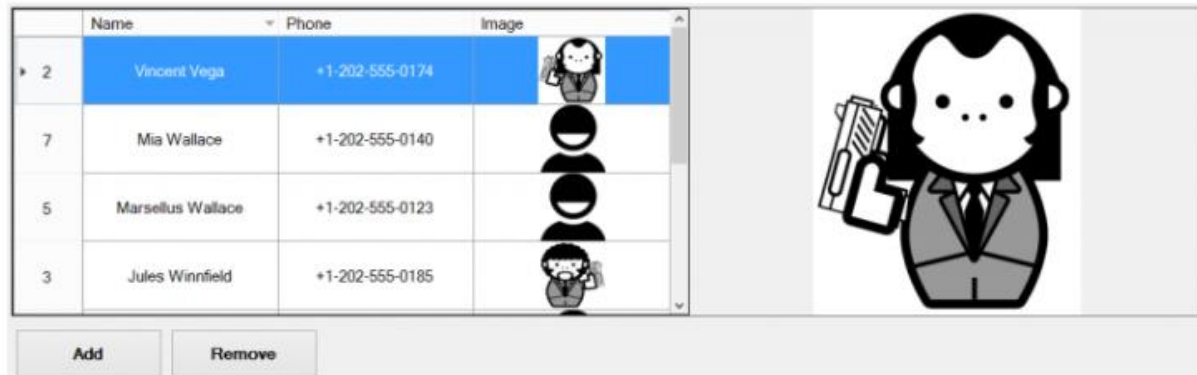
Como el modelo contiene un objeto de base de datos, debe implementar la interfaz `IDisposable` para liberarlo. Sin embargo, si la vida útil del modelo es igual a la vida útil de la aplicación, en la mayoría de los casos no es necesario hacerlo.

Si decide cambiar el tipo de base de datos, la implementación del modelo debe garantizar que la otra parte del *software* permanezca intacta. Por lo tanto, se deben capturar todas las excepciones específicas y se deben manejar todos los errores específicos. Una buena práctica es crear una clase base para las excepciones. Las otras clases de excepción deben heredarla para describir un error más específico. En la aplicación de demostración, `ModelException` es una clase base para las excepciones del modelo.

Paso 3. Agregar formularios y controles de usuario

Las plantillas para formularios y controles de usuario personalizados se pueden crear mediante el Asistente para agregar nuevo elemento en Visual Studio. Este asistente genera una nueva clase para un elemento personalizado. Dicha clase es parcial y, por lo general, consta de dos archivos. Un archivo incluye el contenido que se modifica a través de la ventana Propiedades. Por lo tanto, no se recomienda modificar este archivo manualmente. El otro archivo se puede modificar manualmente para implementar una lógica empresarial necesaria.

Figura 43: La GUI de la aplicación de demostración de la agenda telefónica

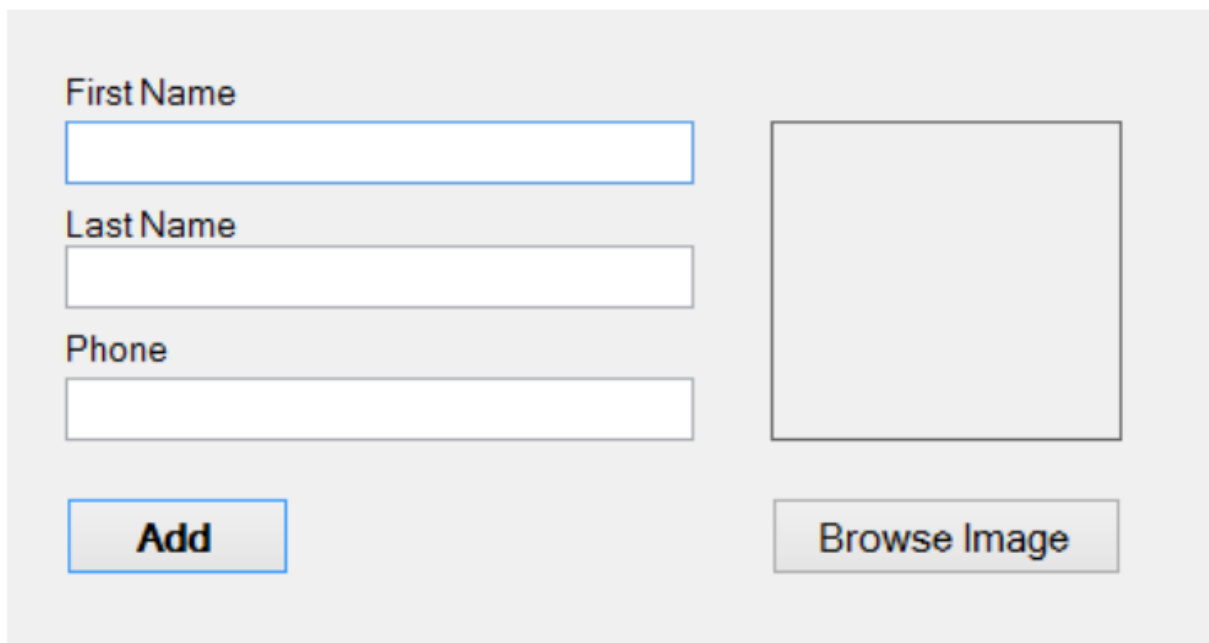


Fuente: (4 steps to create a basic windows forms application|Apriorit, 2016).

El formulario principal contiene el control `ContactView` y dos botones. El control `ContactView` muestra las entradas de contactos tomadas de la base de datos y los botones son necesarios para agregar y eliminar contactos de usuarios.

Por su parte, el control `ContactView` es un control personalizado. La creación de controles personalizados le permite combinar controles existentes en bloques de construcción reutilizables. El control `ContactView` consta de controles `DataGridView` y `PictureBox`, que son controles estándar de WinForms. El primero muestra la información de contacto en la cuadrícula, mientras que el segundo muestra una imagen ampliada del contacto seleccionado en la cuadrícula.

Figura 44: Formulario para agregar un nuevo contacto



The image shows a user interface for adding a new contact. It features three text input fields stacked vertically on the left, labeled 'First Name', 'Last Name', and 'Phone'. To the right of these fields is a large, empty square box. Below the 'First Name' field is a blue-bordered button labeled 'Add'. Below the 'Browse Image' button is a button labeled 'Browse Image'.

Fuente: (4 steps to create a basic windows forms application|Apriorit, 2016).

Paso 4. Implementar la lógica de la GUI

Una vez que se han creado todos los controles y formularios personalizados, se han ajustado sus estilos y se han alineado los controles en los formularios, se debe implementar la lógica para la interacción entre los controles, así como entre la GUI y el modelo. Esta lógica consta de controladores de eventos de control y controladores de eventos de modelo. Una práctica común es heredar las clases de argumentos de eventos de la clase `System.EventArgs` e implementar un evento como una especialización de la plantilla `System.EventHandler`.

Un modelo se actualiza a través de la interfaz gráfica de usuario. Si la actualización es exitosa, el modelo actualiza la interfaz gráfica de usuario a través de los eventos. De lo contrario, los métodos del modelo generan excepciones que deben detectarse y manejarse. El paso más

importante aquí es proporcionar coherencia entre el modelo y los estados de la interfaz gráfica de usuario en caso de errores (4 steps to create a basic windows forms application|Apriorit, 2016).

Herramientas para desarrollo del proyecto

- Visual Basic 2022 (versión local y licencia gratuita).
- Lenguaje de desarrollo principal, integrado con .NET Windows Forms para la interfaz de escritorio.
- Se emplearán licencias definitivas al culminar la etapa de prototipo (Microsoft, 2023 otorgadas por la empresa ya que cuenta con las mismas).
- SQL Server Management Studio (SSMS) 20: herramienta oficial para la administración de bases de datos SQL Server (creación de esquemas, consultas, copias de seguridad, etc.).
- Se instalará de manera local y se configurará para uso en el entorno de pruebas (Kalen et al., 2017).
- Hoja de estilo: uniformiza la apariencia de formularios y controles (botones, cuadros de texto), ofreciendo una experiencia de usuario profesional.
- Servicio SMTP temporal (Gmail) ya que está en un ambiente fuera de la red de la empresa desarrollando la aplicación; una vez puesto en producción, se ligaría con el servidor de correo con el que ya se cuenta.
- Automatiza notificaciones de recuperación de contraseña y alertas de cambios, hasta que se integre el servidor de correo corporativo.

- Archivo .txt del Tribunal Supremo de Elecciones: permite la verificación manual de datos al inicio, con la proyección de integrar un API oficial en el futuro para validar datos directamente en la página, sin tener que ver un txt de manera local (TSE, 2023).

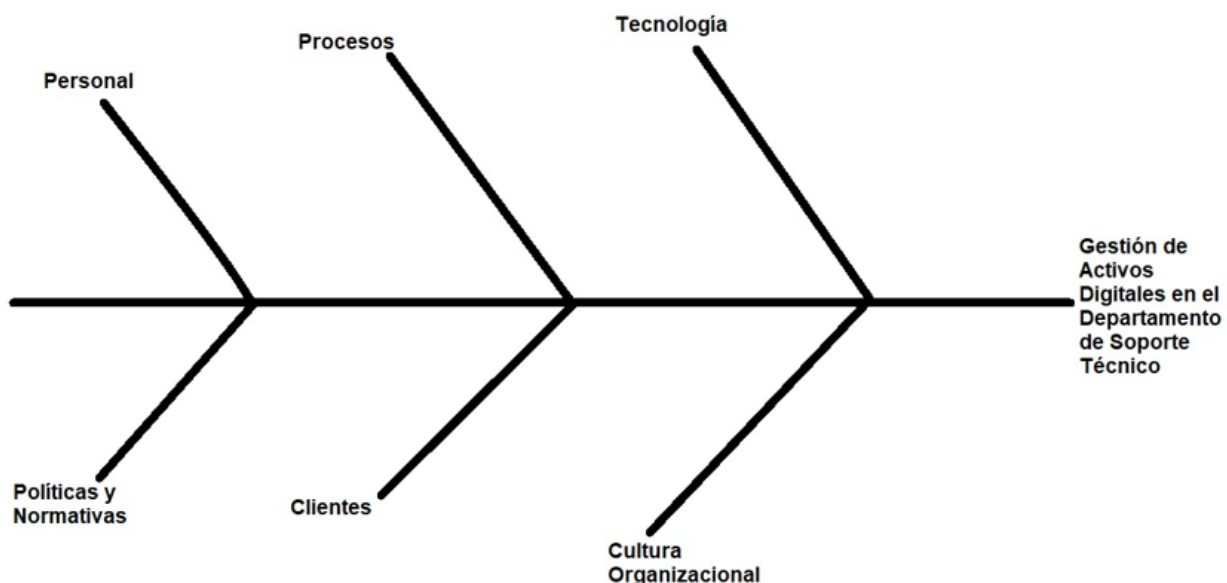
HERRAMIENTAS INGENIERILES

En el Departamento de Soporte Técnico de Instacredit, se pueden utilizar diversas herramientas de planificación para implementar y evaluar un sistema de gestión de activos digitales (DAM).

Diagrama de Ishikawa

En este diagrama, el problema principal se sitúa en la gestión de activos digitales en el Departamento de Soporte Técnico. Las espinas de pescado representan las diferentes categorías de causas que podrían estar contribuyendo al problema (“¿Qué es el Diagrama de Ishikawa? Guía completa | Más Colombia”).

Figura 45: Diagrama de Ishikawa



Fuente: (Argüello Saborío, B. (2025). Diagrama de Ishikawa).

Las espinas de pescado representan las diferentes categorías de causas que podrían estar contribuyendo al problema, las cuales se detallan, a continuación:

- **Personal:** incluye factores relacionados con el personal del Departamento de Soporte Técnico, como falta de capacitación, escasez de personal, rotación de empleados, etc.
- **Procesos:** engloba aspectos relativos a los procesos de gestión de activos digitales, como la falta de procedimientos claros, la ausencia de un sistema formal de seguimiento, la complejidad de los flujos de trabajo, etc.
- **Tecnología:** considera factores tecnológicos que podrían estar afectando la gestión de activos digitales, como la obsolescencia del *software*, la falta de integración entre sistemas, problemas de accesibilidad, etc.

- **Políticas:** incluye políticas internas de la empresa que podrían influir en la gestión de activos digitales, como políticas de seguridad de la información, políticas de respaldo de datos, etc.
- **Clientes:** engloba aspectos relacionados con las necesidades y expectativas de los clientes en cuanto a la gestión de activos digitales, como cambios en los requisitos del cliente, retroalimentación insuficiente, etc.

IDENTIFICACIÓN DE LA EMPRESA

La empresa seleccionada para este estudio es Instacredit, una entidad líder en el sector financiero en Costa Rica. Instacredit se especializa en la provisión de servicios financieros, incluyendo préstamos personales, créditos para vehículos, préstamos hipotecarios, entre otros productos financieros.

Círculo virtuoso de cooperación

- **Valores:** conjunto de principios y creencias que guían el proceder de los seres humanos.
- **Personas:** consideración del factor humano como único recurso “Vivo” que tiene la compañía al vincularse y adaptarse al contexto.

- **Tecnología:** camino de transformación permanente. Debe facilitar la transformación del negocio mediante tecnologías que faciliten la oportunidad de expandir o producir una disrupción que facilite la oportunidad de reinventar los negocios.
- **Clientes:** construir relaciones duraderas y confianza con los clientes permite conectar con ellos, ofreciendo una UX de excelencia y ajustada a sus necesidades.

Antecedentes históricos

Comenzando un nuevo año, un nuevo siglo y milenio, nace la gran empresa, un 26 de abril del año 2000. Abriendo sus puertas en Costa Rica con 23 empleados y 3 sucursales.

Hoy en día cuentan con más de 1200 colaboradores y 53 sucursales en Panamá, Costa Rica, Nicaragua y El Salvador. En estos 24 años han crecido y se han expandido gracias a los clientes y colaboradores.

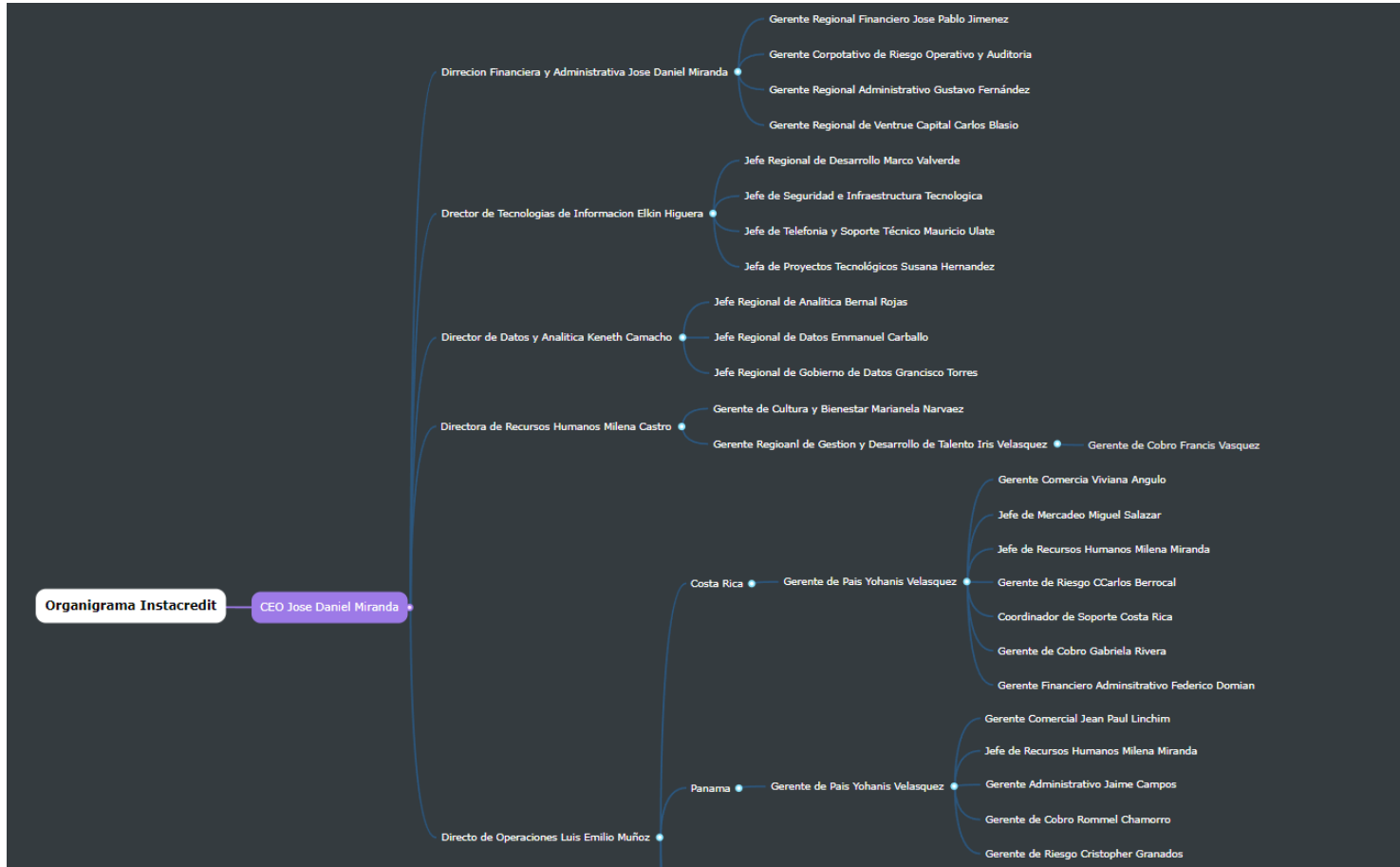
Sus planes de expansión no se han hecho esperar pudiendo hacer realidad en un futuro cercano los corporativos de El Salvador, consolidándose como una empresa transnacional y con una visión de crecimiento y más expansión.

Los colaboradores son su motor y razón de existencia, por esto se preocupan por tener gente motivada y entusiasta que, con su inteligencia e ingenio, hacen crecer este negocio día a día. Todos los miembros de la empresa se encuentran comprometidos con un servicio de calidad que genere satisfacción en su trabajo.

Estructura organizacional

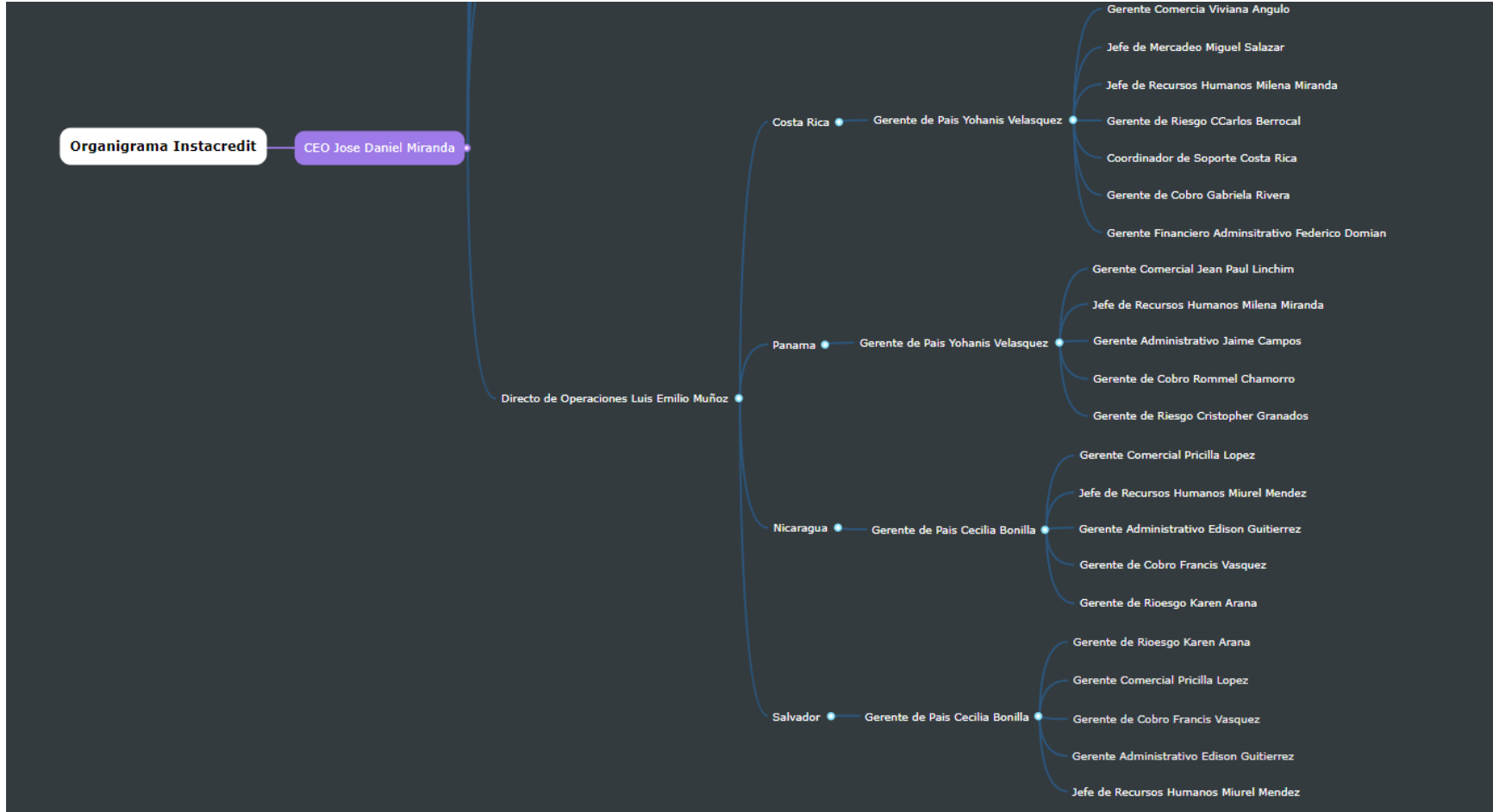
El organigrama de la empresa se muestra, a continuación:

Figura 47: Organigrama de Instacredit S.A.



Fuente: (RRHH Instacredit. (2025). *organigrama de la empresa*. Instacredit).

Figura 48: Organigrama de Instacredit S.A.



Fuente: (RRHH Instacredit. (2025). *organigrama de la empresa*. Instacredit).

Cantidad de empleados

La cantidad de empleados por área se muestra en la siguiente tabla.

Tabla 1: Cantidad de empleados por área

Puesto o Área	Cantidad
Operaciones	850
Tecnologías de la información	150
Datos y analítica	40
Recursos Humanos	50
Financiera y Administrativa	100
Mantenimiento, misceláneos y seguridad	60
Directores	5
Total	1255

Fuente: (RRHH Instacredit. (2025). *Cantidad de colaboradores*. Instacredit).

Tipos de productos

Instacredit, al ser una empresa dedicada al área financiera, puede ofrecer una variedad de servicios que van desde préstamos y créditos hasta servicios financieros complementarios. A continuación, se muestran algunos ejemplos de los tipos de servicios que Instacredit podría ofrecer:

- Préstamos personales: ofrecer préstamos a individuos para cubrir una variedad de necesidades, como gastos médicos, educación, mejoras en el hogar, entre otros.
- Créditos de consumo: proporcionar líneas de crédito renovables para la compra de bienes de consumo, como electrodomésticos, muebles, tecnología, etc.
- Préstamos para vivienda: ofrecer financiamiento para la compra de vivienda, construcción, remodelación o refinanciamiento de hipotecas.
- Préstamos para vehículos: financiar la compra de automóviles nuevos o usados, motocicletas u otros vehículos.
- Servicios de tarjetas de crédito: emitir tarjetas de crédito con diferentes características y beneficios para realizar compras y pagos en establecimientos afiliados.
- Servicios de remesas: facilitar el envío y recepción de remesas internacionales para apoyar a familias y personas que envían dinero a sus países de origen.
- Seguros: ofrecer una variedad de seguros, como seguros de vida, seguros de automóviles, seguros de hogar, entre otros, para proteger a los clientes contra diversos riesgos.
- Servicios de cambio de la moneda: proporcionar servicios de cambio de moneda extranjera para clientes que viajan al extranjero o necesitan realizar transacciones internacionales.
- Servicios de transferencias y pagos: facilitar transferencias de dinero y pagos a través de diversos canales, como transferencias bancarias, pagos en línea, pagos móviles, etc.

- Asesoramiento financiero: brindar asesoramiento financiero personalizado a clientes para ayudarles a administrar sus finanzas de manera efectiva y tomar decisiones informadas.

Mercado de exportación

Costa Rica, Panamá, Nicaragua y Salvador.

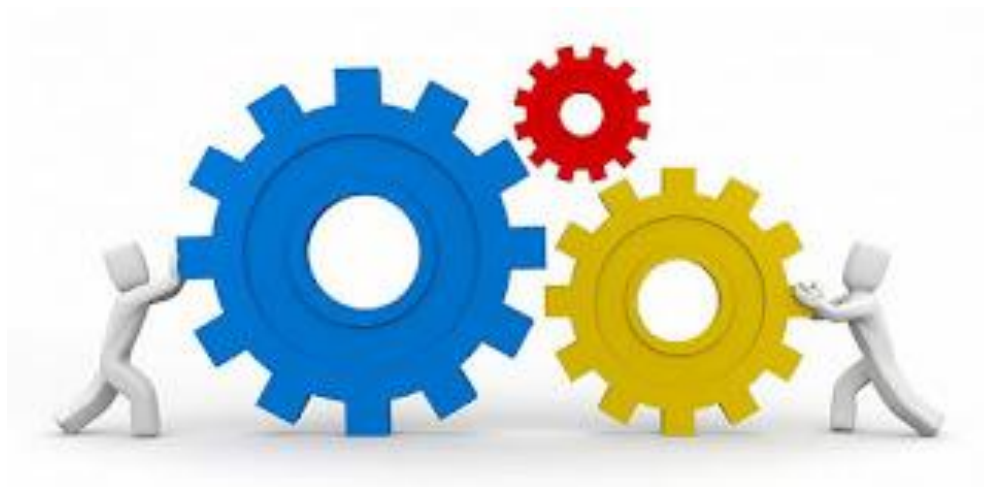
CAPÍTULO III. MARCO METODOLÓGICO

Un marco metodológico es una estructura o conjunto de reglas y principios que guían el proceso de investigación. Es como un mapa que ayuda a los investigadores a planificar y llevar a cabo su estudio de manera organizada y sistemática. Algunos puntos clave son los siguientes:

- ***Diseño de investigación***: el marco metodológico incluye el diseño general de la investigación, es decir, la manera en que se va a recopilar y analizar la información. Por ejemplo, si es un estudio experimental, observacional, de caso, etc.
- ***Métodos de recopilación de datos***: describe las herramientas y técnicas que se utilizarán para recopilar la información. Esto puede incluir encuestas, entrevistas, experimentos, análisis de documentos, entre otros.
- ***Población y muestra***: indica quiénes son los participantes o elementos de estudio (población) y cómo se seleccionará un subconjunto representativo (muestra) para realizar la investigación.
- ***Procedimiento***: detalla los pasos específicos que se seguirán para llevar a cabo la investigación. Esto ayuda a asegurar la consistencia y replicabilidad del estudio.
- ***Instrumentos de medición***: si se utilizan encuestas, tests u otros instrumentos, el marco metodológico describe cómo se desarrollarán y validarán para garantizar su fiabilidad.
- ***Análisis de datos***: explica cómo se procesarán y analizarán los datos recopilados. Esto puede incluir el uso de *software* estadístico, métodos cualitativos o una combinación de ambos, según el tipo de estudio.

- **Ética:** considera aspectos éticos, como la privacidad y el consentimiento de los participantes, para garantizar que la investigación se realice de manera ética y respetuosa (Marco metodológico: Qué es y cómo realizarlo|Ortega, 2024).

Figura 49: Marco metodológico



Fuente: (¿Qué es la metodología? Definición características y tipos?|Pinterest, s.f.).

Importancia del marco metodológico

Contar con un marco metodológico en una investigación es crucial por varias razones, no solo guía el proceso de investigación, sino que también mejora la calidad y la credibilidad de los resultados, permitiendo a otros investigadores comprender, replicar y construir sobre el trabajo realizado.

- **Organización y estructura:** el marco metodológico proporciona una estructura clara y organizada para la investigación. Define los pasos a seguir, desde la

elección de métodos hasta la interpretación de los resultados, lo que ayuda a mantener la coherencia y la lógica en todo el proceso.

- ***Rigurosidad científica***: un marco metodológico sólido contribuye a la rigurosidad científica de la investigación. Establece reglas y normas que aseguran que los métodos utilizados sean apropiados para abordar las preguntas de investigación y que los resultados sean fiables y válidos.
- ***Reproducibilidad y generalización***: al seguir un marco metodológico, se facilita la reproducción del estudio por otros investigadores. Esto es esencial para validar los resultados y generalizar las conclusiones a otras poblaciones o contextos.
- ***Claridad en la planificación***: ayuda a los investigadores a planificar su estudio de manera eficiente. Al definir los métodos, la muestra y los procedimientos por adelantado, se minimiza la posibilidad de errores o malentendidos durante la implementación de la investigación.
- ***Énfasis en la pertinencia***: un buen marco metodológico se adapta a las características específicas de la investigación. Permite a los investigadores seleccionar métodos y técnicas que sean más pertinentes para abordar sus preguntas de investigación, aumentando así la calidad y relevancia de los resultados.
- ***Control de variables***: facilita el control de las variables relevantes. Al establecer criterios específicos para la selección de la muestra, la recopilación de datos y el análisis, el marco metodológico ayuda a minimizar la influencia de variables no deseadas en los resultados.

- **Ética y responsabilidad:** incluye consideraciones éticas que protegen los derechos y el bienestar de los participantes. Esto es esencial para realizar investigaciones de manera responsable y respetuosa.

¿Cómo se desarrolla un marco metodológico?

La elaboración de un marco metodológico implica seguir una serie de pasos para planificar y llevar a cabo una investigación de manera sistemática. A continuación, se detalla una guía general sobre cómo elaborar un marco metodológico:

1. Definición de la investigación:
 - **Objetivos de investigación:** especifica claramente qué se pretende lograr con la investigación. Define los objetivos generales y específicos.
2. Revisión de la literatura:
 - **Marco teórico:** realiza una revisión exhaustiva de la literatura existente sobre un tema. Identifica teorías, conceptos y modelos relevantes que sirvan como base para la investigación.
3. Diseño de investigación: tipo de investigación: es necesario definir si la investigación será descriptiva, exploratoria, experimental, correlacional, etc.
 - **Enfoque:** elige entre un enfoque cuantitativo, cualitativo o mixto, dependiendo de la naturaleza de tus preguntas de investigación.
 - **Conoce más de las características de la investigación mixta.**
4. Población y muestra:

- Población de estudio: identifica el grupo o población que será objeto de investigación.
 - Criterios de inclusión/exclusión: define los criterios para seleccionar participantes (muestra).
5. Instrumentos de recopilación de datos:
- Selección de instrumentos: decide qué herramientas se usan para recopilar datos (encuestas, entrevistas, observaciones, pruebas, etc.).
 - Validación y fiabilidad: si es necesario, validar y verificar la confiabilidad de los instrumentos.
6. Procedimiento:
- Pasos detallados: describe de manera detallada los pasos seguidos para llevar a cabo la investigación, desde la selección de la muestra hasta la recopilación y análisis de datos.
7. Análisis de datos:
- Métodos de análisis: especifica las técnicas estadísticas o cualitativas que utilizarás para analizar los datos.
 - Interpretación de resultados: indica cómo se interpretarán los resultados obtenidos.
8. Aspectos éticos:

- Consideraciones éticas: asegurarse de abordar aspectos éticos, como el consentimiento informado, la confidencialidad y el respeto a los participantes.

9. Plan de trabajo:

- Cronograma: elaborar un cronograma detallado que indique las etapas clave de la investigación y los plazos para cada una.

10. Presupuesto:

- Recursos necesarios: estimar los recursos necesarios, como personal, equipo, material, etc. Si es posible, incluye un presupuesto.

11. Validación del marco metodológico:

- Revisión por expertos: considerar obtener retroalimentación de colegas o expertos en el campo para validar la idoneidad del marco metodológico.

12. Redacción del marco metodológico:

- Estructura clara: redactar el marco metodológico de manera clara y organizada, siguiendo la estructura tradicional de introducción, objetivos, diseño de investigación, población y muestra, etc.

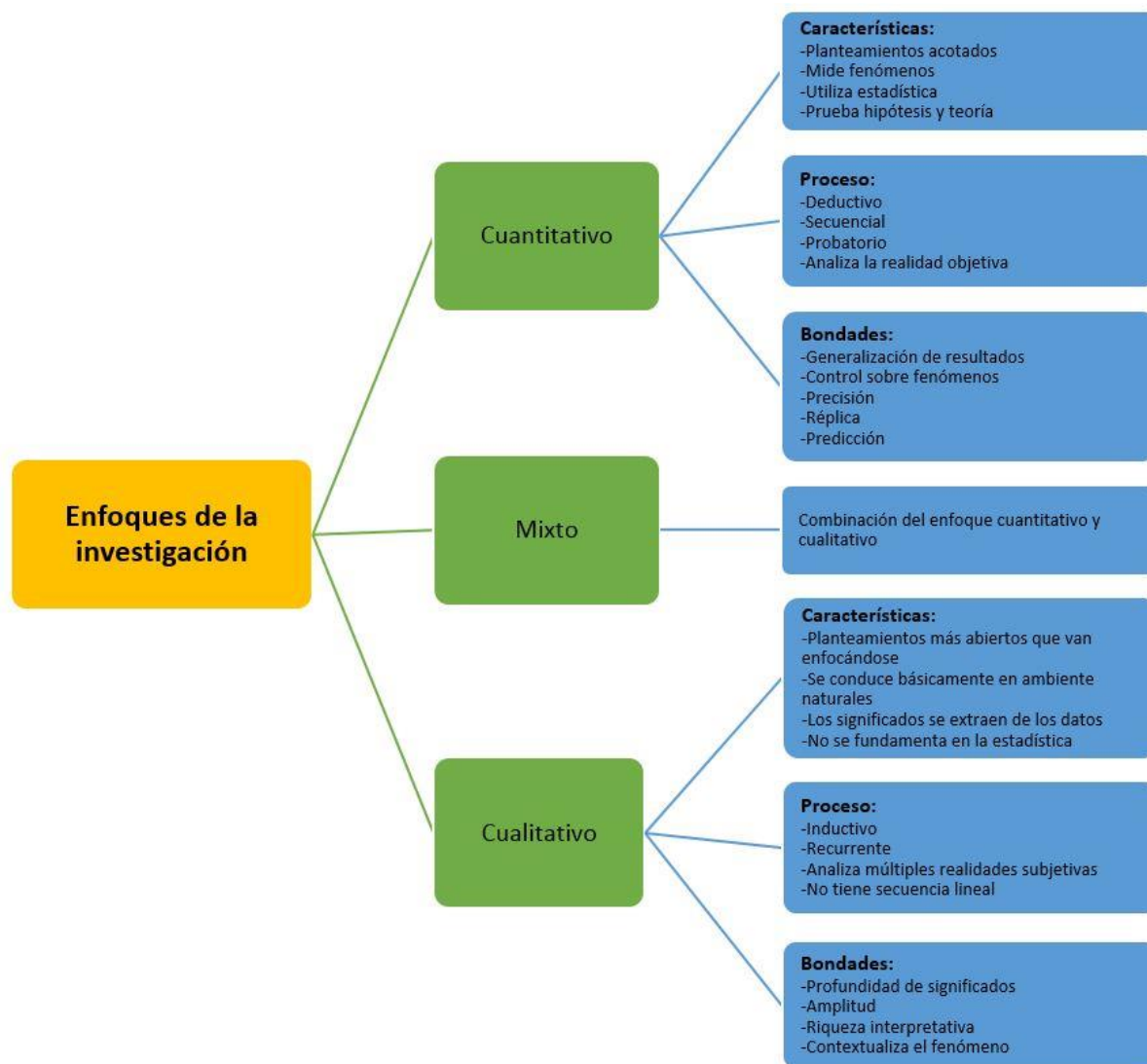
13. Revisión continua:

- Adaptaciones: a medida que se avanza en la investigación, ser flexible para realizar ajustes en el marco metodológico si es necesario.

El marco metodológico proporciona la guía para llevar a cabo la investigación de manera efectiva y rigurosa. Por lo que debe asegurarse de que cada elemento esté claramente definido y

sea coherente con los objetivos del estudio (¿Qué es la metodología? Definición características y tipos?|Pinterest, s.f.).

Figura 50:Enfoque de la investigación



Fuente: (Definición de enfoques de investigación|El portal de la tesis, Ucol.mx).

Tipos de enfoques

¿Cuáles son los enfoques principales de investigación?

A través de la historia de la ciencia, surgieron varias corrientes de pensamiento como el empirismo, el materialismo dialéctico, el positivismo, la fenomenología, el estructuralismo y diversos marcos interpretativos, como el realismo y el constructivismo (Hernández, Fernández y Baptista, 2014).

No obstante, dichas corrientes se concentraron en dos enfoques principales de la investigación: cuantitativo y cualitativo. De ellas se deriva el enfoque mixto.

Figura 51: Enfoque de investigación



Fuente: (Enfoque tipo diseño y método de investigación [Aclarando conceptos])[Samaniego, 2022).

Enfoque cuantitativo

Utiliza métodos y técnicas cuantitativas y tiene que ver con el uso de magnitudes, la observación y medición de las unidades de análisis, el muestreo, así como el tratamiento estadístico. En este enfoque, los procesos se organizan de manera secuencial, rigurosa, de tal manera que se pueda llevar a la comprobación de la hipótesis.

Características:

- Entornos naturalistas.
- Enfoque inductivo.
- Perspectiva holística.
- Subjetividad e interpretación.
- Flexibilidad.

Tipos de investigación cualitativa

Entrevistas en grupo: las entrevistas pueden ser individuales, también grupales y los propósitos son muy diversos.

Entrevistas estructuradas y no estructuradas: se puede contar con una guía de preguntas abiertas o no, así como aplicar preguntas basadas en lo que dice el entrevistador.

Métodos de observación cualitativa: se utiliza para recopilar información sobre los comportamientos no verbales de los sujetos. Tiene la ventaja de que permite estudiar un comportamiento de forma más efectiva al realizar esta acción de manera presencial.

Investigación etnográfica: entre las características que diferencia este tipo de investigación, están:

- La estancia prolongada dentro de la comunidad estudiada para recoger información a través de la observación.
- La elaboración de informes que dan prioridad a la contextualización y la relación interna de los fenómenos que se observan.
- La presentación de interpretaciones que combinan voluntariamente la narrativa.
- La descripción y la conceptualización teórica.

Análisis de redes sociales: el objetivo del análisis de redes es comprender a la comunidad mediante el mapeo de las relaciones que los conectan.

Enfoque cuantitativo

En el caso del enfoque cuantitativo, sí se encuentra enfocado en los números, estadísticas o en procesos que son básicamente relacionados con lo cuantificable. Es un enfoque que está relacionado con procesamiento estadístico o definiciones básicamente cuantificables.

A diferencia del cuantitativo, no es un proceso lineal, sino un proceso “circular”, donde no hay un orden en la secuencia y su proceso es bastante flexible.

Características:

- Sirve para formular leyes y generalizaciones.
- Se resguarda de procedimientos específicos para obtener datos.
- Arroja información sobre una población.
- Se trata de un método objetivo.
- Se vale de la lógica deductiva.

Tipos de investigación cuantitativa

Investigación descriptiva: explica las propiedades, características y rasgos importantes del fenómeno estudiado mediante dinámicas objetivas de observación, análisis y demostración.

Investigación analítica: esto consiste en establecer una relación entre las variables del fenómeno estudiado. En el cual se registran resultados para comprobar o refutar alguna hipótesis previamente establecida.

Investigación experimental: se trata de obtener información mediante un experimento controlado, de forma que se puedan comprender y eventualmente manipular las variables que determinan un fenómeno, para poder establecer cuáles son sus causas y sus efectos.

Enfoque mixto

Aquí se combinan los enfoques cualitativo y cuantitativo, entendiendo que, para lograr la calidad total en el estudio, se requiere complementar los procedimientos de una y otra. Este enfoque puede tener varias secuencias, puesto que lo cuantitativo puede preceder a lo cualitativo o viceversa.

Características:

- En el momento que se quiere validar o corroborar los resultados obtenidos con otros métodos.
- Cada que se necesita utilizar un método para informar a otro.
- Siempre que se quiera analizar continuamente una pregunta de investigación desde diferentes ángulos.

- Cuando se quiere elaborar, clarificar o construir sobre los hallazgos de otros métodos.
- Tan pronto como se quiera desarrollar una teoría sobre un fenómeno de interés y luego ponerla a prueba.
- Si se quiere generalizar los resultados de la investigación cualitativa.

Tipos de investigación mixta

Diseño explicativo secuencial: este diseño implica la recopilación y análisis de datos cuantitativos, seguido de datos cualitativos, priorizando los datos cuantitativos e integrando conclusiones durante la fase de interpretación del estudio.

Diseño exploratorio secuencial: este diseño prioriza la investigación cualitativa, asegurando la recolección y análisis de datos a través de la recolección y análisis de datos cuantitativos, integrando los hallazgos durante la fase de interpretación del estudio.

Estrategia concurrente de triangulación: este diseño utiliza una única fase de recolección y análisis de datos, separando la recolección y análisis de datos cuantitativos y cualitativos, integrando los resultados durante la fase de interpretación del estudio.

Anidado concurrente: este diseño de investigación mixto utiliza una fase de recopilación de datos donde un método predominante se combina con un método menos prioritario, lo que da como resultado la fusión de datos durante la fase de análisis del proyecto.

Método de investigación

Como indica Compilatio (2024), los métodos de investigación son procesos organizados y sistemáticos para obtener información y conocimiento sobre un tema en específico. Se basan

en un conjunto de pasos y técnicas que permiten recolectar, analizar e interpretar datos de manera objetiva y confiable. Los métodos de investigación son utilizados en diversas disciplinas, tanto académicas como científicas, para abordar preguntas de investigación, resolver problemas y llegar a conclusiones basadas en evidencias (Enfoque tipo diseño y método de investigación [Aclarando conceptos][Samaniego, 2022]).

Figura 52: Enfoque cuantitativo



Fuente: (Tipos de investigación cuantitativa: Cuáles son y ejemplos[Narvaez, 2019]).

Figura 53: Enfoque cualitativo



Fuente: (Técnicas de la metodología cualitativa|López, s.f.).

Diseño de la investigación

El diseño de investigación se define como los métodos y técnicas elegidos por un investigador para combinarlos de una manera razonablemente lógica, a fin de que el problema de la investigación sea manejado de manera eficiente.

Así mismo, el diseño es una guía sobre «cómo» llevar a cabo la investigación utilizando una metodología particular. Cada investigador tiene una lista de preguntas que necesitan ser evaluadas.

El bosquejo de cómo debe llevarse a cabo la investigación puede prepararse utilizando el diseño. Por lo tanto, una investigación de mercados se lleva a cabo sobre la base del diseño de esta.

¿Para qué realizar un diseño de investigación?

El diseño de un tema de investigación se utiliza para explicar el tipo de investigación (investigación experimental, encuestas, investigación correlacional, semiexperimental) y también su subtipo (diseño experimental, problema de investigación, estudio de caso descriptivo).

Hay tres etapas principales del diseño de una investigación:

- Recolección
- Medición
- Análisis de datos

Tipos de diseño para una investigación

Un investigador debe tener un claro entendimiento de los diversos tipos de diseño para seleccionar cuál implementar en su estudio. El diseño se puede clasificar ampliamente en diseño cuantitativo y cualitativo.

Diseño de investigación cualitativo: la investigación cualitativa se lleva a cabo en los casos en que se establece una relación entre los datos recopilados y la observación sobre la base de cálculos matemáticos.

Las teorías relacionadas con un fenómeno natural pueden ser probadas o refutadas usando cálculos matemáticos.

Los investigadores se basan en el diseño cualitativo, donde se espera que se concluya «por qué» existe una teoría en particular junto con «qué» tienen que decir los encuestados al respecto.

Diseño cuantitativo: la investigación cuantitativa se lleva a cabo en los casos en los que es importante que un investigador tenga conclusiones estadísticas para recopilar información procesable.

Los números proporcionan una mejor perspectiva para tomar decisiones de negocios importantes. El diseño cuantitativo de la investigación es vital para el crecimiento de cualquier organización porque cualquier conclusión basada en números y análisis resultará ser efectiva para el negocio (Diseño de investigación. Elementos y características|Mugira, 2018).

Fuentes de información

Las fuentes de información son documentos que se consultan para obtener datos de un tema sobre el que se quiere saber más. En una investigación académica, escolar o periodística, conforman el soporte que se usa para producir un escrito u otro tipo de trabajo.

- **Fuentes primarias:** son las más cercanas al evento que se investiga, es decir, con la menor cantidad posible de intermediaciones. Por ejemplo, si se investiga un accidente automovilístico, las fuentes primarias son los testigos directos, es decir, quienes observaron el hecho. Si se investiga un evento histórico, los testimonios directos son una fuente primaria.

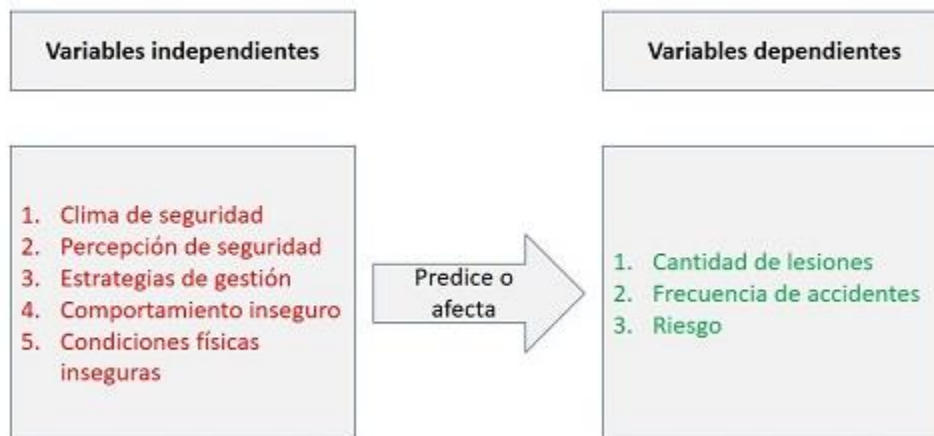
- ***Fuentes secundarias:*** son las que se basan en las primarias y las sintetizan, analizan, interpretan o evalúan. Por ejemplo, si se investiga un accidente, un resumen posterior de los testimonios de los testigos constituye una fuente secundaria. Si se investiga un evento histórico, las fuentes secundarias son aquellos libros que se escribieron tiempo después del suceso y basándose en fuentes primarias o directas.
- ***Fuentes terciarias:*** son las que recopilan y comentan las fuentes primarias o secundarias y, por lo tanto, son una lectura mixta de testimonios e interpretaciones. Por ejemplo, en el caso de un accidente, una fuente terciaria es el archivo policial completo, en el que figuran fotos, testimonios e informes policiales elaborados a partir de estos últimos. Si el evento es un hecho histórico, la fuente terciaria puede ser un manual escolar que se elabora a partir de fuentes secundarias, es decir, libros especializados en el tema.

Fuentes de información confiables

- Indican claramente cuáles son sus propias fuentes. Están realizadas a partir de otras fuentes confiables, de modo que acumulan mayor credibilidad. Por ejemplo, una fuente es confiable si incorpora citas directas o indirectas y si menciona libros, artículos o manuales en las referencias bibliográficas.
- Están realizadas por un autor, una institución o un sitio web especializados en el tema. Llevan la firma de un autor o aparecen en sitios web que son autoridad o referentes en un tema.

- Aplican razonamientos o interpretaciones comprensibles. Exponen sus ideas de manera clara, transparente, coherente, sin ocultar información y sin sacar conclusiones ilógicas.
- Desarrollan el tema de forma objetiva. Abordan los temas de manera seria y rigurosa, generalmente evitando las valoraciones y las opiniones.
- Tienen una fecha reciente. Brindan datos actualizados porque fueron elaboradas hace menos tiempo que otras y eso las vuelve más confiables.
- Evitan el plagio y la repetición. Manejan responsablemente la información, es decir, no repiten lo que otros dicen ni copian la información de terceros.
- Mencionan distintas perspectivas. Abarcan la mayor cantidad posible de puntos de vista, incluso cuando sean contradictorios, para evitar cualquier tipo de sesgo en la investigación.
- Están legitimadas por terceros. Son consideradas confiables por una gran cantidad de investigadores serios y respetados en la materia. (Fuentes de información - Qué son, s.f.)

Figura 55: Tipos de variables, independientes y dependientes



Fuente: (Tipos de variables independientes y dependiente|Tesisdeceroa100.com, s.f.).

Requisitos funcionales

Los requisitos funcionales son especificaciones detalladas de las funciones que un sistema debe realizar para cumplir con las necesidades del usuario. Estos requisitos describen las tareas, procesos y comportamientos que el sistema debe ejecutar.

Por ejemplo, en una aplicación bancaria, un requisito funcional podría ser la capacidad de transferir fondos entre cuentas. Estos requisitos son esenciales para guiar el desarrollo del sistema y asegurar que cumpla con las expectativas del usuario. Además, ayudan a los desarrolladores y diseñadores a entender claramente qué debe hacer el sistema, facilitando la creación de un producto final que satisfaga las necesidades del cliente.

Entre los posibles requerimientos funcionales de un sistema, se incluyen:

- Descripciones de los datos a ser ingresados en el sistema.
- Descripciones de las operaciones a ser realizadas por cada pantalla.
- Descripción de los flujos de trabajo realizados por el sistema.
- Descripción de los reportes del sistema y otras salidas.
- Definición de quien puede ingresar datos en el sistema.
- Cómo el sistema cumplirá los reglamentos y regulaciones de sector o generales que le sean aplicables. (Requerimientos funcionales |Pmoinformatica.com, s.f.)

Requerimientos no funcionales

Los requerimientos no funcionales son criterios que definen cómo debe comportarse un sistema en lugar de describir sus funciones específicas. Estos se centran en aspectos como la seguridad, la usabilidad, el rendimiento, la escalabilidad y la disponibilidad del sistema. Por ejemplo, un requerimiento no funcional puede especificar que un sitio web debe cargar en menos de tres segundos para garantizar una experiencia de usuario óptima.

Además, los requerimientos no funcionales son cruciales para el éxito de un proyecto, ya que ayudan a establecer expectativas claras sobre el rendimiento y la calidad del *software*. Al definir adecuadamente estos requerimientos, los equipos de desarrollo pueden anticiparse a problemas potenciales y garantizar que el *software* no solo cumpla con las especificaciones funcionales, sino que también sea eficiente y confiable en su uso diario. Por lo tanto, incluir ejemplos concretos de requerimientos no funcionales en un artículo puede proporcionar un contexto valioso y fácil de entender para los lectores.

Ejemplos de requerimientos de producto:

- El sistema será desarrollado para las plataformas PC y Macintosh.
- La aplicación debe ser compatible con todas las versiones de Windows, desde Windows 95.
- La aplicación deberá consumir menos de 500 Mb de memoria RAM.
- La aplicación no podrá ocupar más de 2 GB de espacio en disco.
- La nueva aplicación debe manejar fuentes del alfabeto en inglés, idiomas latinos (español, francés, portugués, italiano), arábico y chino.
- La interfaz de usuario será implementada para navegadores web únicamente con HTML5 y JavaScript. (Requerimientos no funcionales|Pmoinformatica.com, s.f.)

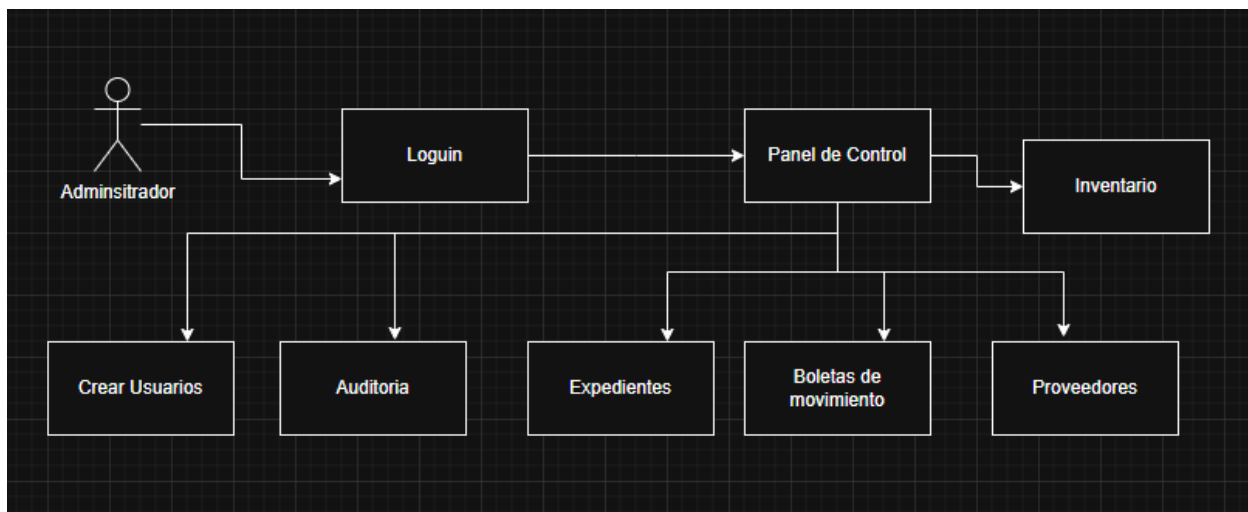
Casos de uso

Los casos de uso se crean para refinar un conjunto de requisitos basados en un rol o tarea. En lugar de la lista tradicional de requisitos que pueden no abordar directamente el uso de la solución, los casos de uso agrupan requisitos comunes basados en el tipo de rol u objetivo. Además, los casos de uso definen lo que los usuarios o roles están haciendo en la solución, un proceso de negocio define cómo realizan estas funciones.:

- **Diagramas de flujo:** que representen las etapas y procesos de la gestión de activos digitales.
- **Casos de uso:** que describan de manera gráfica las interacciones entre los actores (por ejemplo, administrador, colaboradores y auditoría) y el sistema, detallando

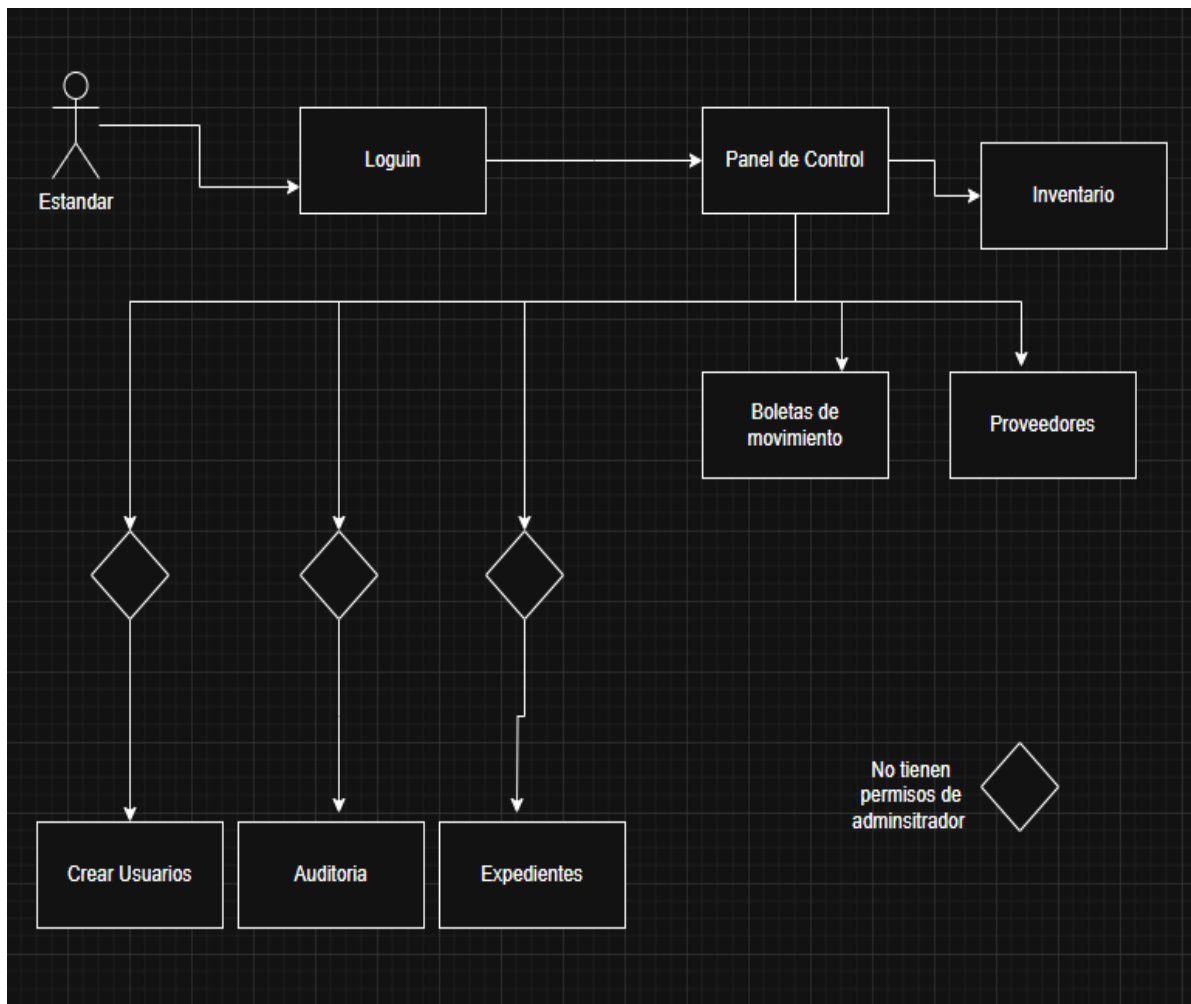
funcionalidades esenciales como la gestión de usuarios, activos, generación de reportes y optimización del sistema. (Bryan Argüello Saborío, 2025)

Figura 56 : Diagrama de flujo de uso Administrador.



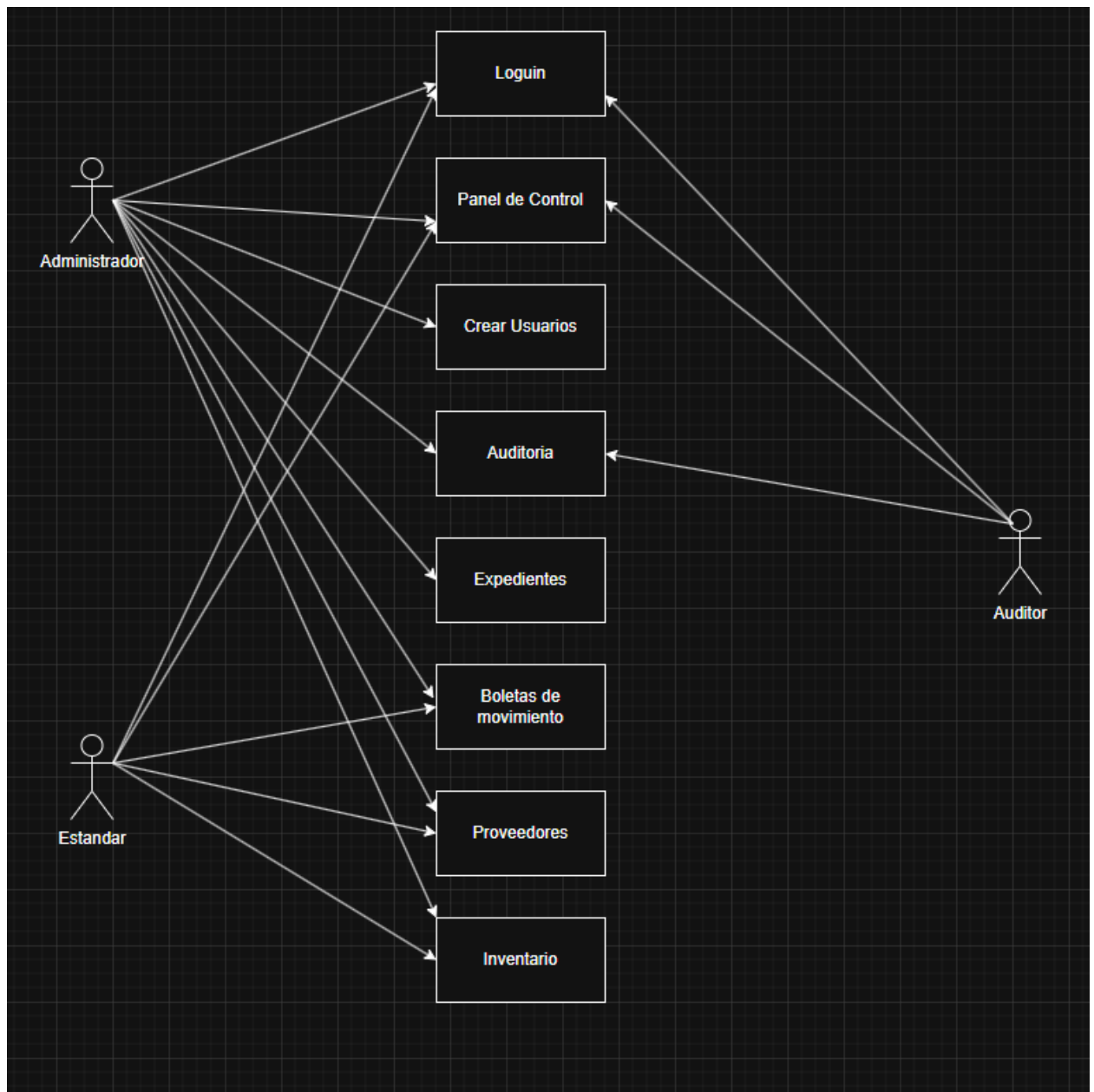
Fuente: (Argüello Saborío, B. (2025). Diagrama de Flujo Administrador).

Figura 57 : Diagrama de flujo estándar



Fuente: Fuente: (Argüello Saborío, B. (2025). Diagrama de Flujo estándar).

Figura 58: Caso de uso.



Fuente: Fuente: (Argüello Saborío, B. (2025). Caso de Uso).

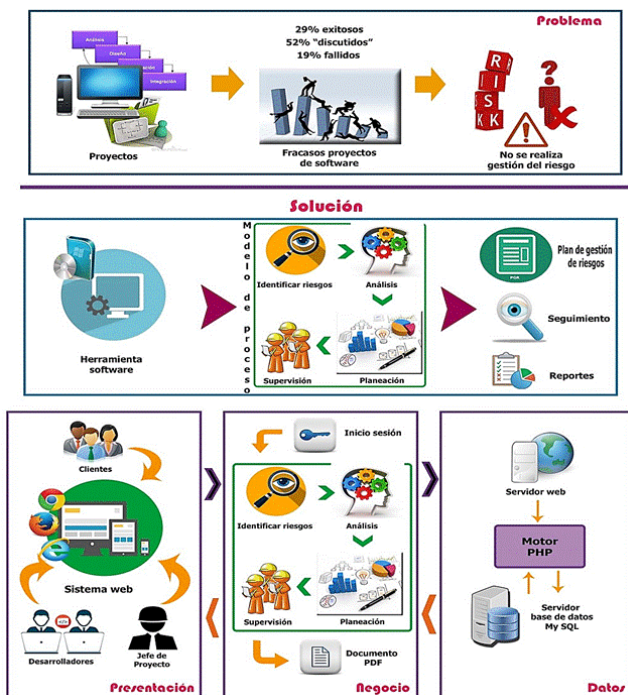
CAPÍTULO IV. ANÁLISIS DE RESULTADOS

Plan de gestión de riesgos

Un plan de gestión de riesgos es un documento en el que se definen los riesgos potenciales que pueden impedir el logro de objetivos en un proyecto, así como las acciones que es preciso implementar para neutralizar esas amenazas, disminuir su impacto negativo o aprovechar posibles oportunidades.

Redactar planes de gestión de riesgos aumenta la probabilidad de éxito de un proyecto o de la gestión de una organización. El objetivo es adelantarse a los acontecimientos y prepararse para afrontar eventos inesperados que pueden surgir en el camino, algunos de ellos con consecuencias negativas, pero otros con interesantes oportunidades para aprovechar (Cómo redactar planes de gestión de riesgos y por qué hacerlo es importante | Escuela Europea de Excelencia, 2022).

Figura 59 : Gestión de riesgo



Fuentes: (Diagrama del problema solución y aspectos tecnológicos|GESTION DE RIESGOS, s.f.).

Componentes principales

La redacción de un plan de gestión de riesgos es un trabajo de equipo. A veces, dependiendo del proyecto, este puede ser multidisciplinar, incluyendo personal de varias áreas relacionadas, como financiera, ingeniería, Alta Dirección o Gestión de Riesgos.

Guía que se basa en siete pasos

Identificar los riesgos

En este primer paso, la participación de personal de diferentes áreas resulta esencial. Las opiniones, experiencias y expectativas del mayor número de personas contribuirán a la elaboración de una lista completa de amenazas probables. Pero la “lluvia de ideas” no es la única metodología para utilizar. También se puede considerar:

Evaluar los riesgos potenciales

De la tarea realizada en el punto uno, surge una lista, en ocasiones extensa, de riesgos y amenazas potenciales. Sin embargo, la experiencia y el simple análisis de la lista indican que no todos son riesgos que representen una preocupación real e inmediata.

Diseñar estrategias para tratar los riesgos

Con una lista categorizada, redactar planes de gestión de riesgos se hace más fácil. En este paso, lo que se hace es diseñar y desarrollar estrategias eficaces para eliminar los riesgos, como prioridad o disminuir su impacto.

Asignar responsabilidades

Con las estrategias aprobadas, es el momento de asignar la responsabilidad de las múltiples tareas que surgieron en el paso anterior. Implementar las estrategias implica planificar actividades, ejecutarlas, verificarlas y comprobar la eficacia integral de la estrategia.

Evaluar el riesgo residual

Pueden subsistir, no obstante, riesgos que no admiten ningún tipo de estrategia. El riesgo residual, el que pasa todos los filtros y aún está ahí, puede en un momento determinado llevar el proyecto a la cancelación cuando es muy alto, no gestionable y con una alta probabilidad de ocurrencia.

Crear un registro de riesgos, documentarlo y compartirlo

Aprovechar la valiosa información que entregan los planes de gestión de riesgos implica adoptar un componente tecnológico que sistematice la gestión, automatice tareas y digitalice sistemas de gestión.

Vigilar los riesgos

Al redactar planes de gestión de riesgos, es preciso tener en cuenta que todos los eventos considerados son susceptibles de cambiar su comportamiento, dejar de existir, aumentar su poder negativo, minimizarlo o convertirse en una gran oportunidad.

También pueden aparecer eventos que no eran considerables en el momento de redactar planes de gestión de riesgos. Por eso, la gestión es continua, cíclica, constante y requiere seguimiento y vigilancia permanente (Cómo redactar planes de gestión de riesgos y por qué hacerlo es importante|Escuela Europea de Excelencia, 2022).

Matriz de riesgos

Una matriz de riesgos es una herramienta de análisis de riesgos que sirve para evaluar la probabilidad y la gravedad del riesgo durante el proceso de planificación del proyecto. Una vez que se haya evaluado la probabilidad y la gravedad de cada riesgo, pueden ubicarse en la matriz para calcular la calificación del impacto de cada riesgo. Estas calificaciones ayudan al equipo a determinar qué prioridad asignar a los riesgos del proyecto y a gestionarlos de manera efectiva (Matriz de riesgos: cómo evaluar los riesgos para lograr el éxito del proyecto [2025]|Asana, 2025).

Tipos de riesgos

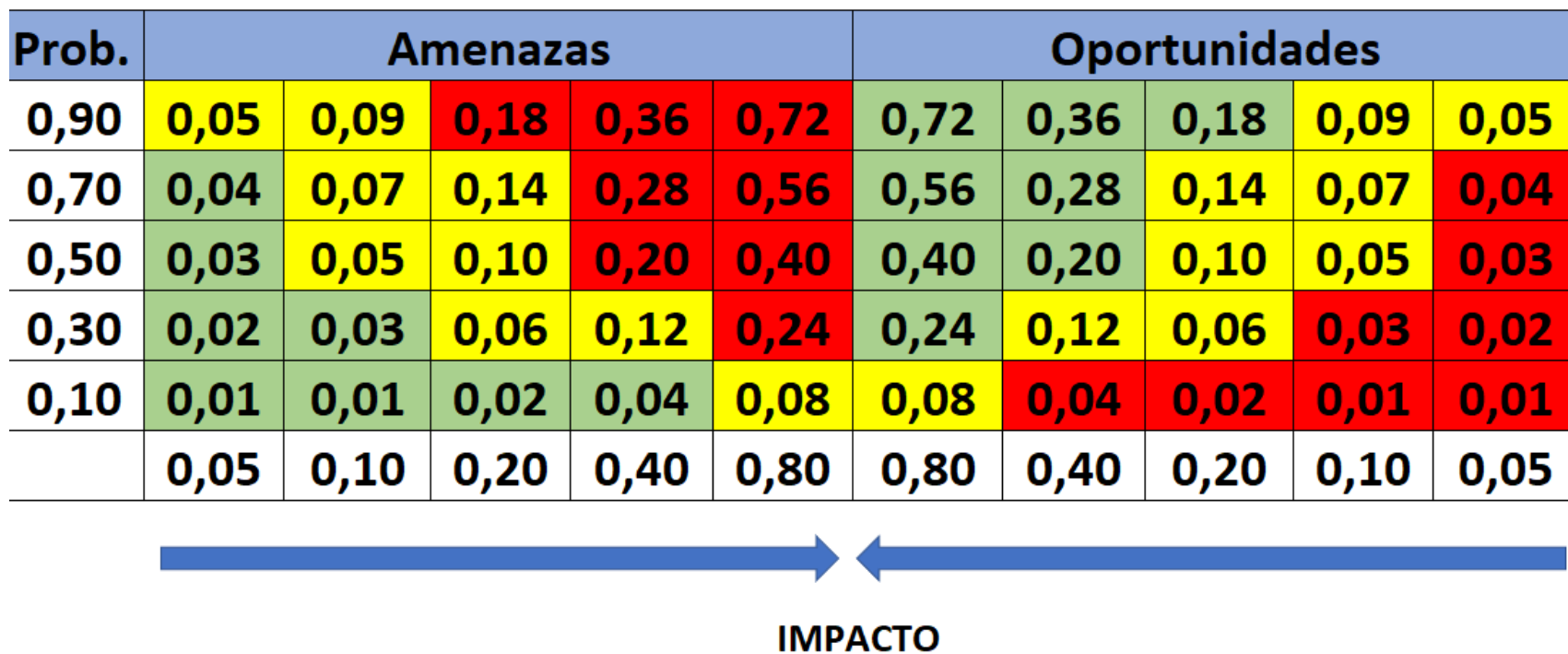
- **Riesgo estratégico:** los riesgos estratégicos implican errores de rendimiento o de decisión, como elegir un proveedor o *software* incorrecto para el proyecto.
- **Riesgo operativo:** los riesgos operativos son errores de proceso o de procedimiento, como una mala planificación o la falta de comunicación entre los equipos.
- **Riesgo financiero:** puede involucrar varios eventos que causen una pérdida de ganancias para la empresa, como los cambios en el mercado, las demandas judiciales o los competidores.
- **Riesgo técnico:** puede incluir cualquier aspecto relacionado con la tecnología de la empresa, como una violación de la seguridad, un corte de energía, la interrupción del servicio de Internet o daños a la propiedad.
- **Riesgo externo:** los riesgos externos están fuera de control, como las inundaciones, los incendios, los desastres naturales o las pandemias.

Figura 60 : Matriz de riesgo

ID	RIESGO	IMPACTO	PROBABILIDAD	NIVEL DE RIESGO	RESPUESTA AL RIESGO	ACCIONES	IMPACTO ESPUES DE RESPUESTA	PROBABILIDAD DESPUES DE RESPUESTA	RIESGO RESIDUAL	COMENTARIOS
RSK01	Riesgos de Seguridad y Protección de Datos	Alto	Probable	Red	Mitigar	Implementar protocolos de cifrado en reposo y en tránsito. Configurar autenticación multifactor. Aplicar control de accesos basados en roles.	Moderado	Poco probable	Bajo	Estas medidas se alinean con la norma ISO 55000 (ISO, 2014) y garantizan la integridad y confidencialidad de la información.
RSK02	Riesgos de Integración y Compatibilidad	Moderado	Probable	Yellow	Mitigar	Diseñar un plan de integración que incluya pruebas de interoperabilidad. Emplear APIs estandarizadas para la comunicación entre sistemas heredados y nuevos.	Moderado	Poco probable	Bajo	Asegura la continuidad del flujo de información, evita inconsistencias y se apoya en buenas prácticas de ingeniería de software (Pressman & Maxim, 2020; Microsoft, 2023).
RSK03	Riesgos de Adopción y Resistencia al Cambio	Moderado	Probable	Yellow	Mitigar	Establecer estrategias de comunicación interna. Realizar talleres de capacitación y sesiones de sensibilización. Destacar beneficios de eficiencia y ahorro.	Moderado	Poco probable	Bajo	Favorece la aceptación del sistema DAM y reduce la incertidumbre del personal. Se alinea con recomendaciones sobre transformación digital (Westerman, Bonnet & McAfee, 2014).

Fuente: (Argüello Saborío, B. (2025). Matriz de Riesgo).

Figura 61 : Mapa de calor de riesgo



Fuente: (Argüello Saborío, B. (2025). Mapa de calor de Riesgo).

Figura 62 : Resultados de mapa de calor

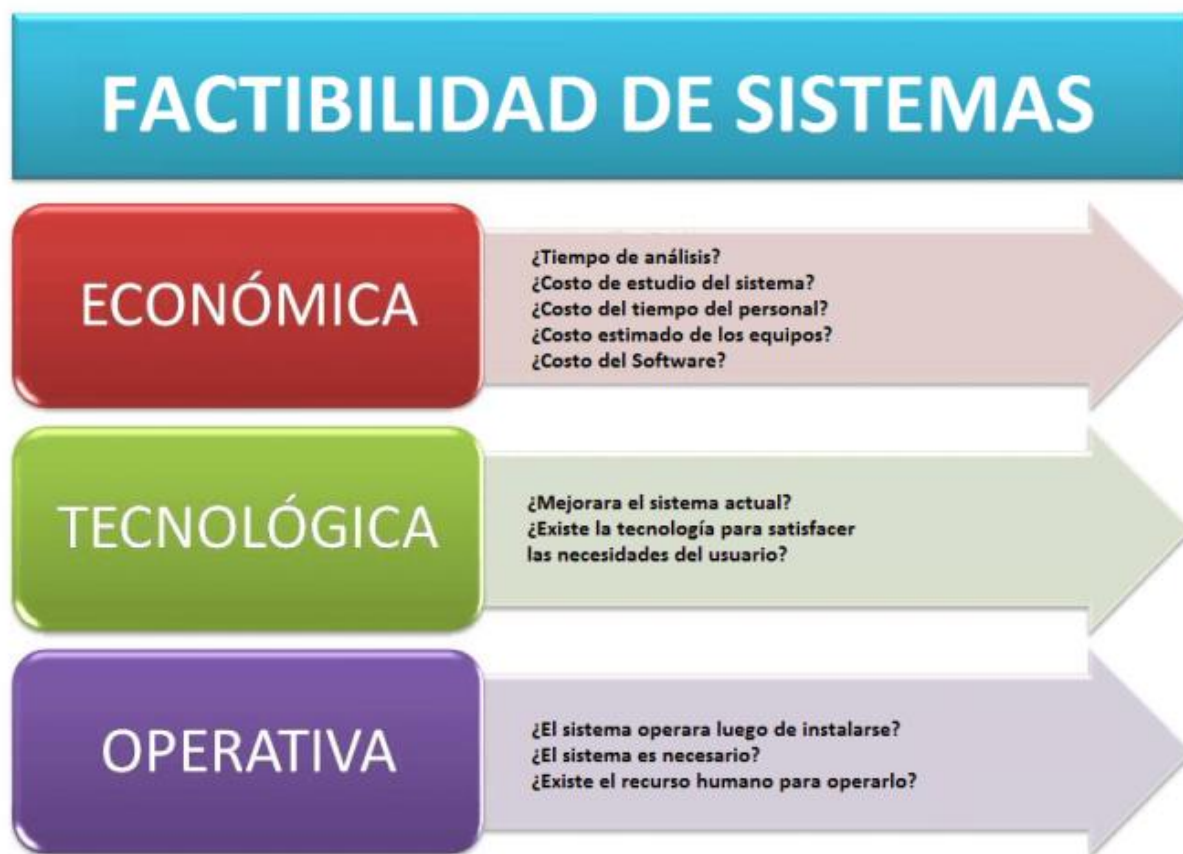
Resultado	
Sin puntuacionSin	0
Sin	
puntuacionExtremadamente	0
Sin puntuacionMuy probale	0
Sin PuntuacionProbable	0
Sin puntuacionPoco Probable	0
Sin puntuacionLigeramente	
Probable	0
Sin puntuacion No aplica	0
Muy Alton Sin Puntuacion	0
Muy AltoExtremadamente	
Probale	Red
Muy AltoMuy Probale	Red
Muy AltoProbale	Red
Muy AltoPoco Probale	Yellow
Muy AltoLigeramente	Yellow
Muy AltoNo aplica	0
AltoSin puntuacion	0
AltoExtremadamente	Red
AltoMuy Probable	Red
AltoProbale	Yellow
AltoPoco Probable	Yellow
AltoLigeramente Probable	Green
AltoNo aplica	0
ModeradoSin puntuacion	0
ModeradoExtremadamnte	
Probable	Red
ModeradoMuy Probale	Yellow
ModeradoProbale	Yellow
ModeradoPocoProbable	Green
ModeradoLigeramente	Green
ModeradoNo aplica	0
Bajosin puntuacion	0
BajoExtremadamente	Yellow
BajoMuy Probale	Yellow
BajoProbable	Green
BajoPoco Probale	Green
BajoLigeramente Probable	Green
Bajo aplica	0
Muy BajoSin puntuacion	0
Muy BajoExtremadamente	
Probale	Yellow
Muy BajoMuy Probale	Green
Muy BajoProbable	Green
Muy BajoPoco Probale	Green
Muy BajoLigeramente	Green

Fuente: (Argüello Saborío, B. (2025). Resultados de Mapa de Calor).

Estudio de factibilidad

Como indica la página *Estudio de factibilidad: ¿por qué es importante para un proyecto?* de Ingeniería VQ (2021), se trata de una evaluación o análisis sobre un nuevo plan, proyecto o método propuesto. Se realiza con la intención de estudiar todos los factores involucrados para reconocer si es viable o no poner en marcha un proyecto o una idea. Además, es importante analizar componentes económicos, técnicos, legales y de programación para determinar la probabilidad de finalizar un proyecto con éxito.

Figura 63: Factibilidad de sistemas



Fuente: (Estudio de Factibilidad|de Luis R Castellanos, 2009).

Factibilidad técnica

Este factor analiza si la infraestructura técnica existente en la empresa Instacredit S.A. puede actuar de manera eficaz y beneficiosa para llevar a cabo el proyecto de gestión de activos digitales que se ha venido desarrollando a lo largo de esta investigación. En relación con el *hardware* requerido para este proyecto, el servidor, estación de trabajo, debe cumplir con requerimientos mínimos, los cuales se muestran en la siguiente tabla:

Tabla 2 : *Factibilidad técnica*

Características	Especificaciones
Sistema Operativo	Windows 10/ Preferible 11 ambos en su versión de 64 bits
Procesador	Intel Opción básica: Intel Core i5-10400 (6 núcleos, 12 hilos) Opción recomendada: Intel Core i5-12400 o superior (6 núcleos, 12 hilos, con mejor eficiencia energética) Opción de alto rendimiento: Intel Core i7-12700 o superior (12 núcleos, 20 hilos, ideal para multitarea con SQL Server) AMD (Ryzen) Opción básica: AMD Ryzen 5 5600G (6 núcleos, 12 hilos, con gráficos integrados) Opción recomendada: AMD Ryzen 5 7600X (6 núcleos, 12 hilos, con alto rendimiento por núcleo) Opción de alto rendimiento: AMD Ryzen 7 7700X o Ryzen 9 7900X (para cargas pesadas en bases de datos y múltiples usuarios)
Almacenamiento	220 GB en SSD o Superior
Memoria Ram	8 GB o Superior
Marca	Cualquiera

Fuente: (Argüello Saborío, B. (2025). Factibilidad técnica).

Al valorar el *hardware* y considerar la configuración mínima necesaria para que el aplicativo pueda ser ejecutado, no es necesaria la inversión de nuevos equipos, dado que se dispone de equipos requeridos para el funcionamiento adecuado de la aplicación de escritorio a desarrollar.

En términos de *software*, se cuenta con todas las aplicaciones requeridas para el desarrollo del proyecto y la empresa cuenta con sus respectivas licencias ya adquiridas, tales

como SQL Server 2022 y Visual Studio 2022. El sistema operativo en los equipos sería Windows 11, lo que sugiere que no es necesario invertir en la adquisición de otros programas.

Factibilidad económica

La factibilidad económica se caracteriza por ser el análisis que se realiza para conocer posibles costos e ingresos que se puede obtener si una empresa saca un producto o servicio al mercado y, en definitiva, si es rentable hacerlo. Es un análisis del tipo costo-beneficio de un determinado proyecto (Te explicamos en qué consiste y cómo realizar la Factibilidad Económica|{Tomas}, 2022).

En este caso, el sistema está siendo desarrollado por un colaborador de la empresa Instacredit S.A., lo que elimina costos adicionales. Además, como el desarrollo del sistema forma parte de una tesis académica, no genera gastos. El servicio no tendrá costo alguno, ya que se trata de un proyecto educativo. Por lo tanto, los costos del desarrollo no se reflejarán en la contabilidad de la empresa Instacredit S.A.

Tabla 3: Factibilidad económica

Características	Especificaciones
Inversión en tecnología	Se aprovechará la infraestructura tecnológica existente (red y servidores) para minimizar costos adicionales.
Licencias	La empresa ya cuenta con las licencias necesarias para el software base, reduciendo el gasto en adquisición de nuevos programas.
Ahorro en costos operativos	Eliminación del uso de papel, reducción en impresión y almacenamiento físico.
Costo de Programación	No se requiere un costo adicional, ya que la programación corre por parte de Bryan Argüello.
Optimización del tiempo	Automatización de procesos administrativos que disminuyen los costos de gestión manual.
Costo de implementación	No será necesario invertir en capacitación externa, dado que la empresa cuenta con un departamento interno de formación. Además, Bryan Argüello se encargará del proceso de capacitación y de la integración del sistema con las API existente para consultas con el Tribunal Supremo de Elecciones.

Retorno de inversión (ROI) Se prevé que la reducción de costos y la mejora en eficiencia operativa compensen la inversión inicial en el mediano plazo.

Fuente: (Argüello Saborío, B. (2025). Factibilidad Económica).

Tabla 4 : Costo del proyecto

Concepto	Descripción	Costo
Infraestructura tecnológica	Se aprovecharán los servidores y red existentes en la empresa.	Sin costo adicional
Licencias de software	La empresa ya cuenta con las licencias necesarias para SQL Server, Visual Basic y otro software relevante.	Sin costo adicional
Costo de desarrollo	El desarrollo corre por cuenta del estudiante (Bryan Argüello) como parte de su tesis.	Sin costo adicional
Capacitación del personal	Será realizada internamente por el departamento de capacitación de la empresa y el propio desarrollador.	Sin costo adicional
Implementación de seguridad	La empresa ya cuenta con certificados SSL adicionales o autenticación multifactor de terceros.	Sin costo adicional
Hardware adicional (opcional)	Se aprovecharán los servidores y equipos existentes en la empresa.	Sin costo adicional
Integración con API empresariales	Se utilizarán API que ya forman parte de los sistemas existentes de la empresa.	Sin costo adicional
Mantenimiento y soporte técnico	El mantenimiento, soporte y seguimiento del proyecto corre por parte de desarrollador.	Sin costo adicional
Desarrollo	Por un estudiante de Ingeniería Informática	Sin costo adicional
Proyecto	Parte de una tesis académica	No se cobrará por el servicio
Contabilidad	No generará gastos en la Empresa	No reflejados

Fuente: (Argüello Saborío, B. (2025). Costo del Proyecto).

Tabla 5: Costo del proyecto

Concepto	Descripción	Costo Aproximado (USD)
Infraestructura tecnológica	Adquisición de un servidor físico adecuado para alojar la base de datos SQL Server: Dell PowerEdge T40	\$ 950,00
Licencias de software (1 año)	Incluye SQL Server Standard Edition, Visual Studio Professional y Windows Server Standard Edition.	\$ 2 500,00
Costo de desarrollo	Desarrollo del software personalizado por un programador independiente.	\$ 2 600,00
Capacitación del personal	Cursos y sesiones prácticas para 10 personas, impartidos por equipo de capacitación de la empresa.	\$ 800,00
Implementación de seguridad	Adquisición de certificado SSL, configuración de autenticación multifactor y firewall.	\$ 300,00
Hardware adicional	2 laptops con procesador Intel Core i5 de 13ª generación, 16 GB de RAM y 512 GB SSD.	\$ 1 600,00
Integración con API	Desarrollo de conectores y pruebas para integrar API institucionales (ej. padrón electoral).	\$ 400,00
Mantenimiento y soporte técnico	Soporte anual, actualizaciones y respaldo técnico.	\$ 600,00
Contabilidad / Control financiero	Adaptación de procesos internos para la integración contable del sistema.	\$ 300,00
Total Estimado		\$ 10 050,00

Fuente: (Argüello Saborío, B. (2025). Costo del Proyecto).

Factibilidad operativa

La factibilidad operativa consiste en el análisis de los recursos productivos, incluidos los humanos, necesarios para la realización de un proyecto económico. Por tanto, la factibilidad operativa se centra en los procesos de la empresa, a diferencia de otras como la técnica (recursos técnicos) o la financiera (recursos financieros).

Etapas para analizar la factibilidad operativa

Como ya se ha mencionado, se debe conocer a fondo los procesos productivos de la empresa o las fases de realización de los diferentes servicios. Esto ayudará a analizar la factibilidad operativa.

- En primer lugar, conviene mantener una reunión con los responsables de producción y el personal implicado. Hay que tener claro qué se busca, con qué se cuenta y dónde se quiere llegar.
- Una vez se ha planificado, el director de producción deberá preparar un informe y realizar un seguimiento de este. Este debe ser lo más detallado posible, incluyendo todos los recursos productivos y operativos necesarios.
- Por último, se debe enviar este informe a la Gerencia para que estudie su viabilidad. Si finalmente es aprobado, se pondrá en marcha estableciendo unos periodos de control para evitar posibles desviaciones. (Factibilidad Operativa|Economipedia.com, s.f.)

Descripción del proyecto:

Desarrollo de un sistema de gestión de activos digitales, utilizando SQL Server, Visual Studio 2022 en lenguaje .net Windows forms.

- **Funcionalidades:** creación de usuarios con rol y validación de este contra permisos previamente otorgados por el administrador del sistema, gestión de boletas de movimientos de activos, proveedores, inventarios, expedientes, reportes de auditoría y monitoreo del estado de salud de las tablas y de la conexión con el servidor.
- **Contexto:** parte de una tesis académica, sin costos adicionales para la empresa Instacredit S.A.

Objetivos del proyecto

- Reducir tiempos de procesos del personal y costo en materia prima para imprimir documentos y mantenerlos almacenados.
- Mejorar la trazabilidad de todas las operaciones realizadas en el sistema.
- Proporcionar un sistema seguro y fácil de usar.

Recursos necesarios

Hardware

- Servidor para alojar la base de datos.
- Computadoras para los usuarios del sistema.

Software

- SQL Server.
- Visual Studio 2022.
- .Net Framework.

Evaluación de riesgo

- **Dependencia del desarrollador:** riesgo de falta de disponibilidad del único desarrollador.
- **Tiempo de implementación:** ajustar el desarrollo e implementación al cronograma académico.
- **Problema de programación:** problemas de versiones.

Evaluación de la viabilidad operativa

La empresa Instacredit S.A. no incurre en costos adicionales significativos.

Tabla 6 : Factibilidad operativa

Aspecto	Descripción
Facilidad de Uso	El sistema contará con una interfaz intuitiva y accesible, asegurando que los usuarios puedan operar sin dificultad. Se realizarán capacitaciones internas para facilitar la adopción del conocimiento del uso.
Resistencia al Cambio	Se implementarán estrategias de comunicación y talleres de sensibilización para minimizar la resistencia al cambio entre los empleados.
Adaptabilidad del Usuario	Se considera un período de transición en el que se brindará soporte a los usuarios, garantizando que se adapten progresivamente al nuevo sistema sin afectar sus funciones diarias.
Sostenibilidad y Actualización	El sistema está diseñado con tecnologías escalables que permiten futuras actualizaciones sin afectar su funcionamiento. Se establecerán planes de mantenimiento preventivo para evitar obsolescencia rápida (Esto luego de la implementación).
Descripción del Proyecto	Desarrollo de un Sistema de Gestión de Activos Digitales (DAM) utilizando tecnologías compatibles con la infraestructura de Instacredit S.A.
Funcionalidades Clave	Control centralizado de activos, generación de reportes, autenticación de usuarios, gestión de permisos y trazabilidad de documentos.
Recursos Necesarios	Se utilizará la infraestructura tecnológica existente (servidores y red). No se requieren costos adicionales en licencias.
Recursos Humanos	Un estudiante de Ingeniería Informática con experiencia en .NET y bases de datos, apoyado por el departamento de proyectos de la empresa, esto aprobado por el director de tecnología de Instacredit.
Evaluación de Riesgos	Dependencia del desarrollador (se documentará el código y se capacitará a personal interno), posibles retrasos en la implementación (se ajustará a los tiempos académicos y empresariales), compatibilidad con otros sistemas.
Evaluación de Viabilidad Operativa	El sistema mejorará la eficiencia operativa del área de soporte técnico sin costos adicionales significativos, asegurando una implementación exitosa.

Fuente: Fuente: (Argüello Saborío, B. (2025). Factibilidad Operativa).

Figura 64: Estudio de factibilidad



Fuente: (Tipos de Factibilidad|My first document, s.f.)

Metodología utilizada

En el desarrollo del programa de gestión de activos digitales para la empresa Instacredit S.A., se adoptó la metodología Kaban. Según se menciona en (¿Qué es la metodología Kanban y cómo funciona?|Martins, 2025), permite:

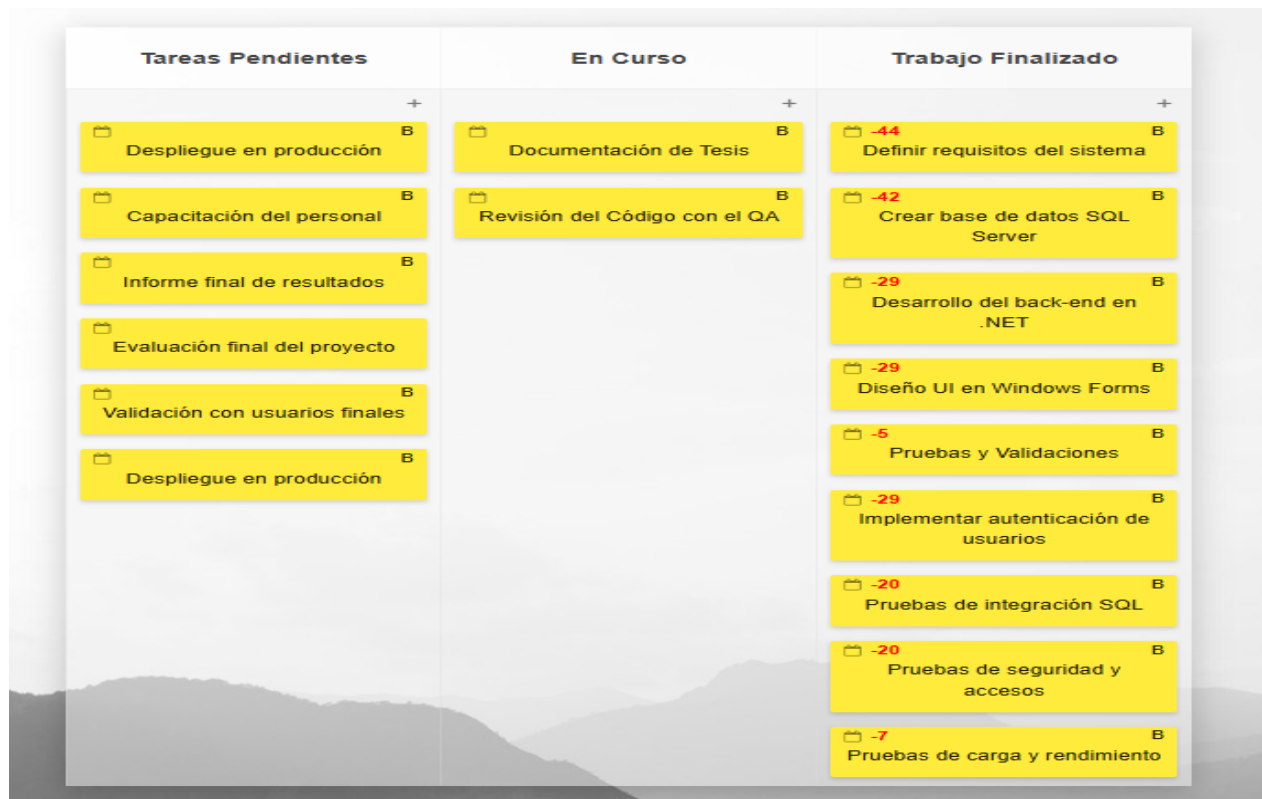
- **Visualizar el trabajo:** una de las principales ventajas de Kanban es que puede visualizar cómo el trabajo “avanza” a través de las etapas. Una tarjeta Kanban de tarea comenzará su viaje en el lado izquierdo del tablero y, a medida que el equipo trabaja en ella, recorrerá lentamente las siguientes etapas hasta que aterrice en la columna Finalizadas.
- **Limitar el trabajo en curso:** como metodología ágil, Kanban se centra en un principio de entrega temprana, lo que implica que las tareas deben moverse rápidamente de una columna a otra en lugar de estancarse en un estado ambiguo de “trabajo en progreso” (wip).
- **Gestionar el flujo de trabajo:** gestionar y mejorar el flujo de trabajo permitirá controlar el tiempo predestinado para el trabajo y así poder reducir el tiempo de entrega (el tiempo que pasa entre el inicio de una tarea hasta que llega a la columna Finalizadas del tablero Kanban) y garantizar que está entregando tareas o enviando nuevos productos mientras siguen siendo relevantes.
- **Implementar políticas de procesos explícitas:** las políticas del proceso deben guiar al equipo en la implementación de la metodología Kanban. Además, se debe

alentar a todos en el equipo a participar e innovar las políticas Kanban, tal como se establece en el cuarto principio básico de Kanban: impulsar el liderazgo en todos los niveles.

- **Implementar ciclos de comentarios:** en Kanban, se necesita recopilar comentarios de dos grupos distintos: los clientes y el equipo.
 - Recopila comentarios de los clientes sobre la calidad y eficacia de la solución que produjo el equipo. ¿Fue el producto adecuado? ¿Hubo algún problema? En el caso de que haya surgido algún problema, como errores en un código o cualquier otro defecto del producto, revisar el flujo Kanban y agregar más tiempo para la revisión, los ajustes y la evaluación.
 - Realiza consultas frecuentes con el equipo sobre el proceso de ejecución de un marco Kanban. ¿Cómo se sienten con los resultados? Aquí tiene otra oportunidad para fomentar el liderazgo en todos los niveles y mejorar las políticas de procesos del equipo.
- **Mejorar colaborando y evolucionar experimentando:** en esencia, Kanban se trata de una mejora continua. Sin embargo, también significa que otros sistemas podrían funcionar bien junto con Kanban. Ya sea Scrum o alguna otra metodología, debe estar siempre dispuesto a colaborar, experimentar y desarrollar los procesos si es necesario.

Con este enfoque, se logra una administración dinámica y flexible que se adaptaría a las necesidades del proyecto y garantiza el cumplimiento de los objetivos.

Figura 65: Cronograma de tareas y control método Kanban



Fuente: (Argüello Saborío, B. (2025). Cronograma de Tareas y Control Método Kanban).

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Evaluación de situación actual: la investigación reveló que la falta de un sistema de gestión de activos digitales en el área de soporte técnico de Instacredit S.A. ha generado ineficiencias significativas, incluyendo la duplicación de activos, la pérdida de documentos y problemas de trazabilidad. Esta evaluación confirma la necesidad de una solución tecnológica que digitalice los procesos manuales.
- Definición de requerimientos técnicos y operativos: a partir del análisis realizado, se establecieron los requerimientos necesarios para la implementación del sistema de gestión de activos digitales, garantizando una solución efectiva para reducir la duplicación de activos, mejorar el control documental y optimizar la gestión de inventarios.
- Diseño del sistema: se desarrollaron modelos UML que representan los procesos del sistema, incluyendo la gestión de usuarios, activos y reportes, lo que facilita su comprensión y futura implementación.
- Implementación y validación: se construyó un prototipo funcional utilizando SQL Server y un entorno de desarrollo en Visual Basic, lenguaje .net Windows form, validando su capacidad para automatizar procesos, mejorar la trazabilidad y proporcionar mayor seguridad en la gestión de activos.
- Impacto y beneficios esperados: la implementación del sistema ASSETPRO permitirá una mejora significativa en la eficiencia operativa del área de soporte técnico, garantizando una gestión segura y centralizada de los activos digitales, reduciendo costos y errores humanos.

Recomendaciones

- Integración progresiva: se recomienda conectar ASSETPRO con otras plataformas de la empresa, como Active Directory, Servidor de correo interno, aplicación de monitoreo ZABBIX, para mejorar la interoperabilidad y ampliar su funcionalidad a futuro.
- Capacitación adaptativa: aunque no se desarrolle un manual de usuario formal, es fundamental realizar capacitaciones periódicas para el personal, asegurando una adopción efectiva del sistema y maximizando su uso.
- Monitoreo y mejora continua: se deben establecer revisiones trimestrales para evaluar el desempeño del sistema, detectar posibles fallas y proponer mejoras con base en datos reales.
- Fortalecimiento de la seguridad: se recomienda mantener actualizados los mecanismos de cifrado de contraseñas y otros protocolos de seguridad para garantizar la integridad y protección de los datos.
- Escalabilidad del sistema: se debe considerar la posibilidad de expandir ASSETPRO a otras áreas de la empresa, asegurando su adaptabilidad a nuevas necesidades operativas.

CAPÍTULO VI. ANÁLISIS DE REQUERIMIENTOS

Análisis de requerimientos

Como se menciona en *¿Qué es la gestión de requisitos?* de Ibm.com (2024), la gestión de requisitos es un conjunto de técnicas para documentar, analizar, priorizar y acordar requisitos, a fin de que los equipos de ingeniería siempre tengan requisitos actualizados y aprobados. Los requisitos claros, concisos y sin errores ayudan a los equipos de ingeniería a detectar errores de forma temprana, lo que reduce los costos y riesgos del proyecto.

El propósito de la gestión de requisitos es garantizar que los objetivos de desarrollo de *software* y productos se cumplan con éxito. Además, proporciona una forma de evitar errores mediante el seguimiento de los cambios en los requisitos. Esta técnica también fomenta la comunicación con los *stakeholders* desde el inicio de un proyecto a lo largo del ciclo de vida de la ingeniería.

Para el prototipo de gestión de activos digitales de la empresa Instacredit S.A., se decidió que había que enfocarse en estos tres requerimientos:

- Requerimientos de usuario.
- Requerimientos funcionales.
- Requerimientos no funcionales.

Requerimientos de usuario

Los requerimientos de usuario son un documento donde se describe “qué” debe hacer el sistema en términos no técnicos y debe ser lo más detallado posible para evitar ambigüedades.

En la mayoría de las empresas, también llaman a los requerimientos de usuario (RU) o especificación de requerimientos de usuario.

Requerimientos funcionales

Los requisitos funcionales son especificaciones detalladas de las funciones que un sistema debe realizar para cumplir con las necesidades del usuario. Estos requisitos describen las tareas, procesos y comportamientos que el sistema debe ejecutar (Requerimientos funcionales |Pmoinformatica.com, s.f.).

Requerimientos no funcionales

Los requerimientos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Suelen presentar dificultades en su definición, dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir (Requerimientos no funcionales|Pmoinformatica.com, s.f.).

Tabla 7 : Requerimiento de usuario RUM01

Responsable: Bryan Argüello Saborío			
Id:	RUM01	Prioridad:	Alta
Descripción:			
Diseñar un sistema integral para la gestión y control de activos digitales que permita administrar, registrar y monitorear tanto activos físicos como digitales del área de soporte técnico. El sistema deberá soportar la actualización en tiempo real, manejo de inventario y generación de reportes para facilitar la toma de decisiones.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			

Criterios de Aceptación:
Visualización y actualización en tiempo real del inventario de activos.
Generación de reportes descargables en formatos PDF.
Alertas automáticas ante niveles críticos de stock o incidencias en la trazabilidad.

Tabla 7: Requerimiento generado por el estudiante para la empresa Instacredit S.A.

cantidad de datos dentro de cada módulo.

Tabla 8: Requerimiento de usuario RUM02

Responsable: Bryan Argüello Saborío			
Id:	RUM02	Prioridad:	Alta
Descripción:			
Desarrollar un módulo de autenticación (Login) que reconozca y diferencie roles de usuario (por ejemplo, administrador y estándar) para el acceso seguro al sistema de activos digitales. Deberá incluir validación de credenciales y mecanismos de recuperación de contraseña.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			
Autenticación segura basada en una base de datos encriptada.			
Funcionalidad de recuperación de contraseña mediante correo electrónico.			

Tabla 8: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para ingreso al sistema.

Tabla 9: Requerimiento de usuario RUM03

Responsable: Bryan Argüello Saborío			
Id:	RUM03	Prioridad:	Alta
Descripción:			
Módulo para recuperar contraseña, si al usuario se le olvida, lo hace otra vez de su usuario generado.			

Fuente:
Área de Soporte Técnico – Instacredit
Dependencias:
Ninguna
Criterios de Aceptación:
validación de contraseña segura.
validación de usuarios existente en la base de datos.

Tabla 9 Requerimiento generado por el estudiante para la empresa Instacredit S.A. para recuperar la contraseña para ingresar al sistema.

Tabla 10: Requerimiento de usuario RUM04

Responsable: Bryan Argüello Saborío			
Id:	RUM04	Prioridad:	Alta
Descripción:			
Cambio de contraseña de usuario al primer inicio de sesión cuando se hace la apertura por primera vez o cuando lleve 90 días con la misma contraseña.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			
Funcionalidad para importar y exportar datos de inventario.			
Reportes automáticos y alertas de stock mínimo.			
Interfaz para ajustes manuales y auditorías.			

Tabla 10: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para cambiar la contraseña en el primer ingreso al sistema para usuarios nuevos.

Tabla 11: Requerimiento de usuario RUM05

Responsable: Bryan Argüello Saborío			
Id:	RUM05	Prioridad:	Alta
Descripción:			
Desarrollar un módulo de panel de control, para que maneje la identificación de los roles existentes y así permita utilizar todos los módulos o niegue por falta de permisos.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			
Visualización y actualización en tiempo real del inventario de activos.			
Generación de reportes descargables en formatos PDF para auditoría			
Crear usuarios con diferentes Roles.			
Envío de boletas con automatización de actualización de inventarios y envío de boletas por correo.			
Alertas automáticas ante niveles críticos de stock o incidencias en la trazabilidad.			

Tabla 11: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para ingreso al panel de control del sistema para usuarios (Identifica si es Administrador o Estándar).

Tabla 12: Requerimiento de usuario RUM06

Responsable: Bryan Argüello Saborío			
Id:	RUM06	Prioridad:	Alta
Descripción:			
Desarrollar un módulo para el registro de usuarios que permita crear, editar e inhabilitar perfiles, asignar roles y gestionar permisos específicos. Deberá incluir validación de campos obligatorios y, opcionalmente, verificación mediante correo electrónico.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			

Ninguna
Criterios de Aceptación:
Formularios con validación de datos y confirmación de registro.
Asignación de campos solo letras.
Asignación de campos solo números.
Asignación de roles configurable por el administrador.
Integración de notificaciones para nuevos registros o modificaciones.

Tabla 12: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para crear usuarios para ingreso al sistema (Solo Administrador puede crear usuarios).

Tabla 13: Requerimiento de usuario RUM07

Responsable: Bryan Argüello Saborío			
Id:	RUM07	Prioridad:	Alta
Descripción:			
Desarrollar un módulo para que el control de auditoría registre todas las actividades relevantes de los usuarios (inicio de sesión, modificaciones, eliminaciones) para garantizar la trazabilidad, seguridad y facilitar auditorías internas del sistema.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			
Registro seguro y detallado de las actividades.			
Funcionalidad de consulta y generación de reportes de auditoría.			
Almacenamiento cifrado de la información.			

Tabla 13: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para generar reportes de auditorías tales como bitácoras de usuarios, inventarios y movimientos de boletas (Solo Administrador puede descargar estos reportes).

Tabla 14: Requerimiento de usuario RUM08

Responsable: Bryan Argüello Saborío			
Id:	RUM08	Prioridad:	Alta
Descripción:			
Desarrollar un módulo para el registro de expedientes de usuarios internos de soporte. Deberá permitir almacenar datos de amonestaciones, fecha, dar seguimiento, actualizar, buscar, etc.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			
Carga de datos a una tabla en la base de datos.			
Actualizar datos en tabla en base de datos.			
Mostrar los usuarios que se tengan registrados en la tabla de usuarios.			

Tabla 14: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para crear expedientes de seguimiento para equipo de soporte (Solo Administrador puede crear Expedientes).

Tabla 15: Requerimiento de usuario RUM09

Responsable: Bryan Argüello Saborío			
Id:	RUM09	Prioridad:	Alta
Descripción:			
Desarrollar un módulo para el registro y control del inventario de activos digitales y físicos, que permita realizar conteos periódicos, ajustes de stock y generar reportes de auditoría interna. Se deberán incorporar métodos de conteo cíclico y alertas de reposición.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			

Mejor control de inventarios.
Inventarios centralizados lo que hace más efectivo el tiempo del equipo.

Tabla 15: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para ingresar, consultar, actualizar, revisar y buscar equipos en el inventario que se maneja en Instacredit.

Tabla 16: Requerimiento de usuario RUM10

Responsable: Bryan Argüello Saborío			
Id:	RUM10	Prioridad:	Alta
Descripción:			
Desarrollar un módulo para el registro de proveedores, que permita almacenar y gestionar información de contacto, historial de transacciones y evaluación de calidad de los servicios ofrecidos para la adquisición o mantenimiento de activos digitales.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			
Trazabilidad de tiempos de vencimiento 90-60-30 días.			
Datos claros de los contratos que se tiene para reportes de averías.			
Facilidad para actualizar datos de contacto y transacciones.			

Tabla 16: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para llevar un mejor control de los proveedores.

Tabla 17: Requerimiento de usuario RUM11

Responsable: Bryan Argüello Saborío			
Id:	RUM11	Prioridad:	Alta
Descripción:			
Desarrollar un módulo para crear nuevos proveedores de servicios para el departamento de soporte técnico			
Fuente:			
Área de Soporte Técnico – Instacredit			

Dependencias:
Ninguna
Criterios de Aceptación:
Trazabilidad de tiempos de vencimiento 90-60-30 días.
Datos claros de los contratos que se tiene para reportes de averías.
Facilidad para actualizar datos de contacto y transacciones.

Tabla 17 Requerimiento generado por el estudiante para la empresa Instacredit S.A. para crear nuevos proveedores que se adquieran por parte de la empresa.

Tabla 18: Requerimiento de usuario RUM12

Responsable: Bryan Argüello Saborío			
Id:	RUM12	Prioridad:	Alta
Descripción:			
Desarrollar un módulo para llevar un mejor control de los movimientos de las boletas que se le asignan al personal y, a su vez, se actualiza directamente el inventario.			
Fuente:			
Área de Soporte Técnico – Instacredit			
Dependencias:			
Ninguna			
Criterios de Aceptación:			
Actualización automática de equipos en el inventario cuando se ejecuta.			
Realizar un mejor control de las boletas de inventario por correo.			

Tabla 18: Requerimiento generado por el estudiante para la empresa Instacredit S.A. para el envío de boletas de manera digital a los usuarios finales.

Requerimientos funcionales

A continuación, se especifica la lista de requisitos funcionales que el sistema debe cumplir para asegurar un correcto funcionamiento.

Tabla 19: Requerimiento funcional RFM01

Responsable: Bryan Argüello Saborío			
Id:	RFM01	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, inhabilitar y modificar los registros de usuarios. El administrador es el responsable de gestionar el acceso y los permisos de cada cuenta.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 19: Requerimiento funcional para la gestión de usuarios, contemplando altas, bajas, actualizaciones y modificaciones de sus datos y permisos.

Tabla 20: Requerimiento funcional RFM02

Responsable: Bryan Argüello Saborío			
Id:	RFM02	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, inhabilitar y modificar los registros de proveedores. El administrador y los usuarios son los encargados de realizar estas acciones y de mantener la información actualizada.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 20: Requerimiento funcional para la gestión de proveedores, permitiendo el control total de sus datos dentro de la plataforma.

Tabla 21: Requerimiento funcional RFM03

Responsable: Bryan Argüello Saborío			
Id:	RFM03	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, inhabilitar y modificar los registros de activos (por ejemplo, equipos, licencias de software, dispositivos). El administrador y los usuarios son responsables de su correcta administración.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 21: Requerimiento funcional para la gestión integral de activos, que contempla altas, modificaciones y bajas lógicas (inhabilitaciones).

Tabla 22: Requerimiento funcional RFM04

Responsable: Bryan Argüello Saborío			
Id:	RFM04	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar y modificar los registros de boletas de movimiento. El administrador y los usuarios controlan las transferencias, ingresos, reasignaciones y bajas de activos.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 22: Requerimiento funcional para manejar el flujo de activos, documentando cada movimiento dentro de la organización.

Tabla 23: Requerimiento funcional RFM22

Responsable: Bryan Argüello Saborío			
Id:	RFM05	Prioridad:	Alta
Descripción:			
El sistema permite agregar, actualizar, inhabilitar y modificar los registros de expedientes de soporte. Tanto el administrador como los usuarios con permisos podrán documentar incidencias y acciones correctivas.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 23: Requerimiento funcional que facilita la gestión de expedientes, registrando toda la información asociada a solicitudes y tickets de soporte.

Tabla 24: Requerimiento funcional RFM06

Responsable: Bryan Argüello Saborío			
Id:	RFM06	Prioridad:	Alta
Descripción:			
El sistema permite generar y descargar reportes relacionados con activos, movimientos, usuarios y proveedores. El administrador define los criterios de búsqueda y el formato de los reportes.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 24: Requerimiento funcional para la creación de reportes, facilitando la toma de decisiones y la auditoría interna.

Tabla 25: Requerimiento funcional RFM07

Responsable: Bryan Argüello Saborío			
Id:	RFM07	Prioridad:	Alta
Descripción:			
El sistema mantiene un registro de auditoría (quién, qué y cuándo se realizaron cambios). El administrador puede visualizar y filtrar los eventos registrados para identificar acciones indebidas o no autorizadas.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 25: Requerimiento funcional que garantiza la trazabilidad de las acciones realizadas en la plataforma.

Tabla 26: Requerimiento funcional RFM08

Responsable: Bryan Argüello Saborío			
Id:	RFM08	Prioridad:	Alta
Descripción:			
El sistema permite gestionar roles y permisos de los usuarios, definiendo qué acciones pueden realizar dentro de la aplicación. El administrador controla la asignación de roles y la modificación de privilegios.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 26: Requerimiento funcional para la asignación de roles y permisos, asegurando la seguridad y el control de accesos en la plataforma.

Tabla 27: Requerimiento funcional RFM09

Responsable: Bryan Argüello Saborío			
Id:	RFM09	Prioridad:	Alta
Descripción:			
El sistema registra inicios y cierres de sesión, mostrando la fecha y hora de cada evento. El administrador es responsable de revisar la bitácora para garantizar la transparencia y seguridad en el acceso.			
Fuente:			
Coordinador de Soporte Costa Rica			
Dependencias:			
Ninguna			

Tabla 27: Requerimiento funcional que registra las sesiones de los usuarios, permitiendo un control sobre la actividad y la seguridad de la plataforma.

Requerimientos no funcionales

Tabla 28: Requerimiento no funcional RNFM01

Responsable: Bryan Argüello Saborío			
Id:	RNFM001	Prioridad:	Alta
Descripción:			
El sistema debe garantizar la seguridad de la información almacenada y transmitida, encriptando credenciales de acceso y protegiendo los datos sensibles de proveedores, usuarios y activos.			
Fuente:			
Coordinador de Soporte Costa Rica			

Tabla 28: Requerimiento no funcional que enfatiza la seguridad del sistema, asegurando la confidencialidad de los datos y la protección de la información sensible.

Tabla 29: Requerimiento no funcional RNFM02

Responsable: Bryan Argüello Saborío			
Id:	RNFM002	Prioridad:	Alta
Descripción:			
El sistema debe ofrecer alta disponibilidad, permitiendo el acceso a la plataforma las 24 horas del día, los 7 días de la semana, con un tiempo de inactividad programado no mayor a 2 horas al mes para tareas de mantenimiento.			
Fuente:			
Coordinador de Soporte Costa Rica			

Tabla 29: Requerimiento no funcional orientado a la disponibilidad continua, asegurando la mínima interrupción del servicio y la productividad de los usuarios.

Tabla 30: Requerimiento no funcional RNFM03

Responsable: Bryan Argüello Saborío			
Id:	RNFM003	Prioridad:	Alta
Descripción:			
El sistema debe ser escalable, de manera que soporte el incremento en el número de usuarios, activos y registros sin afectar el rendimiento o la respuesta del servicio.			
Fuente:			
Coordinador de Soporte Costa Rica			

Tabla 30: Requerimiento no funcional que se centra en la escalabilidad del sistema, garantizando un desempeño estable aun cuando aumente la demanda.

Tabla 31: Requerimiento no funcional RNFM04

Responsable: Bryan Argüello Saborío			
Id:	RNFM004	Prioridad:	Alta
Descripción:			
El sistema debe ser intuitivo y fácil de usar, proporcionando una interfaz clara y accesible para todo el personal. Las funciones principales deben ser identificables rápidamente, reduciendo la curva de aprendizaje.			
Fuente:			
Coordinador de Soporte Costa Rica			

Tabla 31: Requerimiento no funcional que subraya la usabilidad, promoviendo una experiencia de usuario satisfactoria y reduciendo errores de operación.

Tabla 32: Requerimiento no funcional RNFM05

Responsable: Bryan Argüello Saborío			
Id:	RNFM005	Prioridad:	Alta
Descripción:			
El sistema debe contar con registros de auditoría que rastreen los cambios realizados en la plataforma (quién, cuándo y qué modificó). Estos logs deben ser almacenados de forma segura y accesible solo para el administrador y personal autorizado.			
Fuente:			
Coordinador de Soporte Costa Rica			

Tabla 32: Requerimiento no funcional centrado en la trazabilidad y la responsabilidad de los cambios en el sistema, reforzando la seguridad y el control interno.

Modelos UML

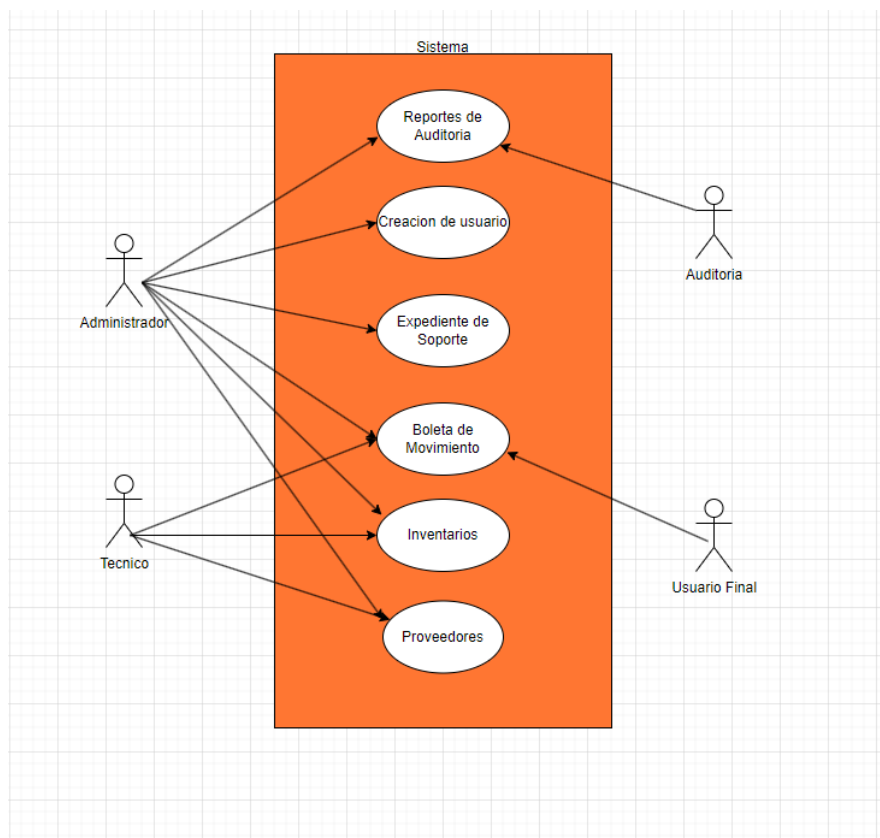
Casos de uso

A continuación, se detallan los actores involucrados en el sistema:

Administrador: es la persona que se encarga de mantener un registro exacto de los usuarios que podrán utilizar sistema.

Usuario: es la persona que puede agregar, eliminar o modificar información (boletas de movimiento, proveedores, inventarios), aunque no tiene los permisos para crear usuarios, inhabilitar usuarios, asignar roles a otros usuarios, descargar reportes e ingresar al panel de expedientes.

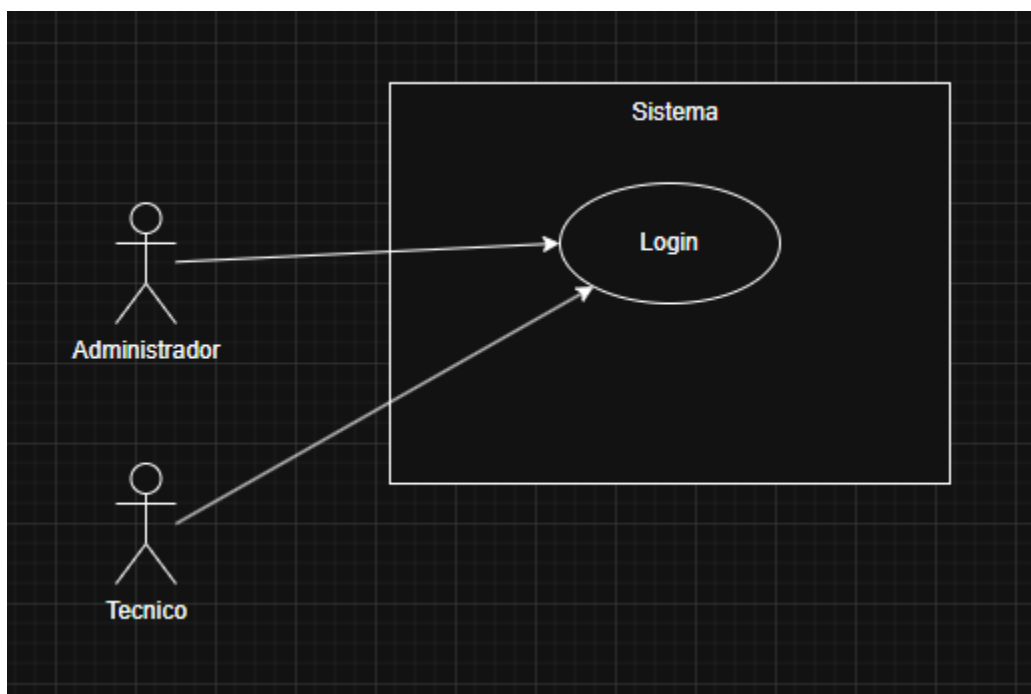
Figura 66: Diagrama general de casos de uso



La figura 66 muestra el diagrama general del programa de gestión de activos digitales.

Caso de uso 1. Login

Figura 67: Diagrama de caso de uso - Login



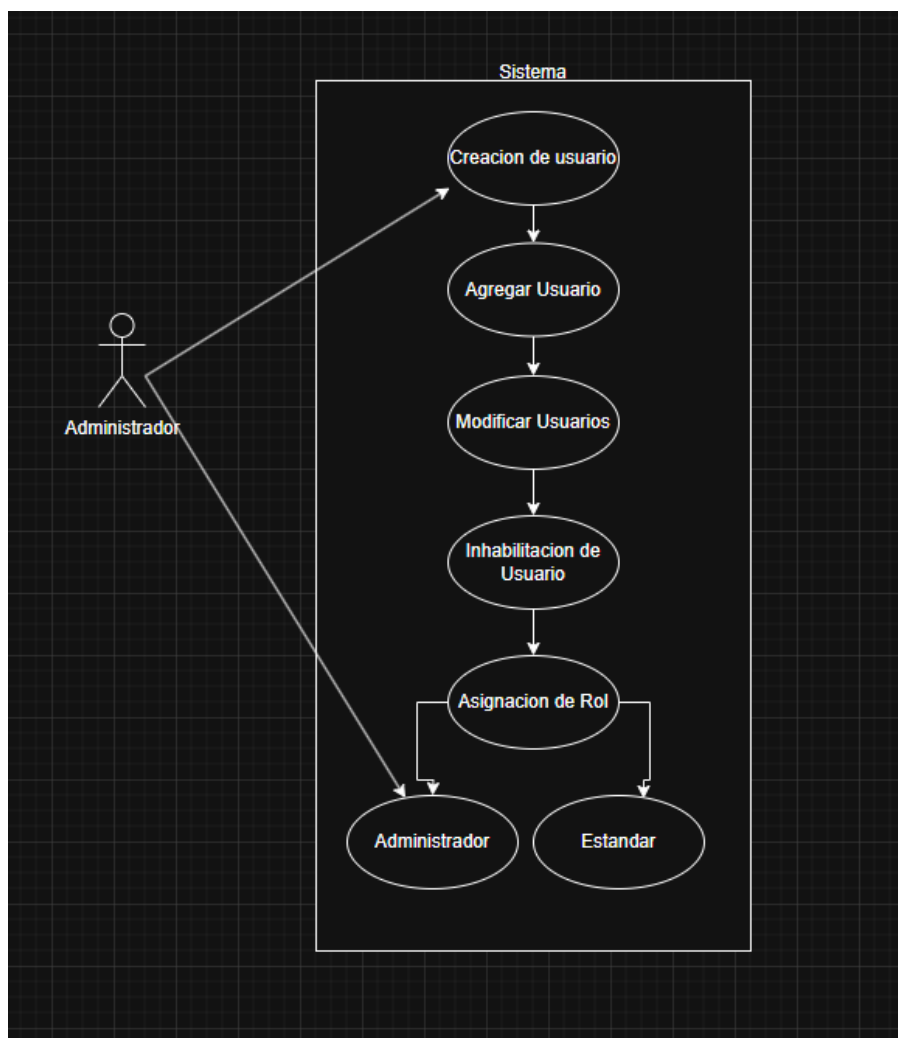
La figura 67 muestra el diagrama de caso de uso que pertenece al Login del sistema de gestión de activos digitales de InstacreditS.A.

Tabla 33: Login

Casos de uso	Login
Autor(a)	Bryan Argüello Saborío
Actores	- Administrador - Usuario
Observaciones	- Ingreso de credenciales - Cierre de sesión
Descripción	1. El usuario (Administrador/Usuario) accede a la pantalla de Login. 2. Ingresa su usuario generado y contraseña. 3. Presiona el botón “Iniciar Sesión”. 4. El sistema valida las credenciales. 5. Si son correctas, redirige al panel principal según el rol. 6. Para salir, se presiona “Cerrar Sesión”.
Mensajes de error	- Si la contraseña es incorrecta, se muestra un mensaje de error. - Si el usuario no existe, se muestra un mensaje de error.

2. Caso de uso: registro de usuarios

Figura 68 : Diagrama de caso de uso de registro de usuarios



La Figura 68 del diagrama de caso de uso pertenece al registro de usuarios del sistema de gestión de activos digitales de Instacredit.

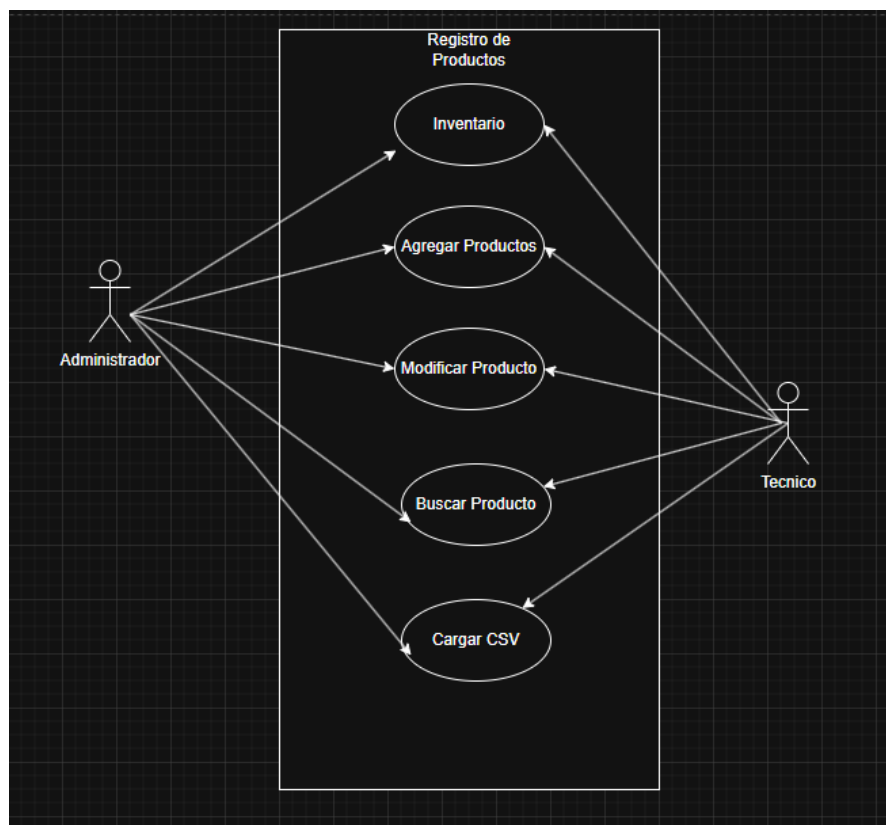
Tabla 34: Registro de usuario

Casos de uso	Registro de Usuarios
Autor(a)	Bryan Argüello Saborío
Actores	- Administrador - Sistema
Observaciones	- Lista los datos de usuarios - Agrega, modifica o inhabilita usuarios

Descripción	1. El administrador accede a la sección “Usuarios”. 2. Ingresa los datos requeridos (correo, contraseña, rol). 3. Presiona “Guardar”. 4. El sistema valida que no exista otro usuario con el mismo correo. 5. Si es válido, registra al nuevo usuario en la base de datos.
Mensajes de error	- Si el correo ya existe, se muestra un mensaje de error. - Si faltan campos obligatorios, el sistema indica cuáles faltan.

3. Caso de uso: registro de activos

Figura 69: Diagrama de caso de uso de registro de activos



La figura 69 del diagrama de caso de uso pertenece al registro de activos del sistema de gestión de activos digitales de Instacredit.

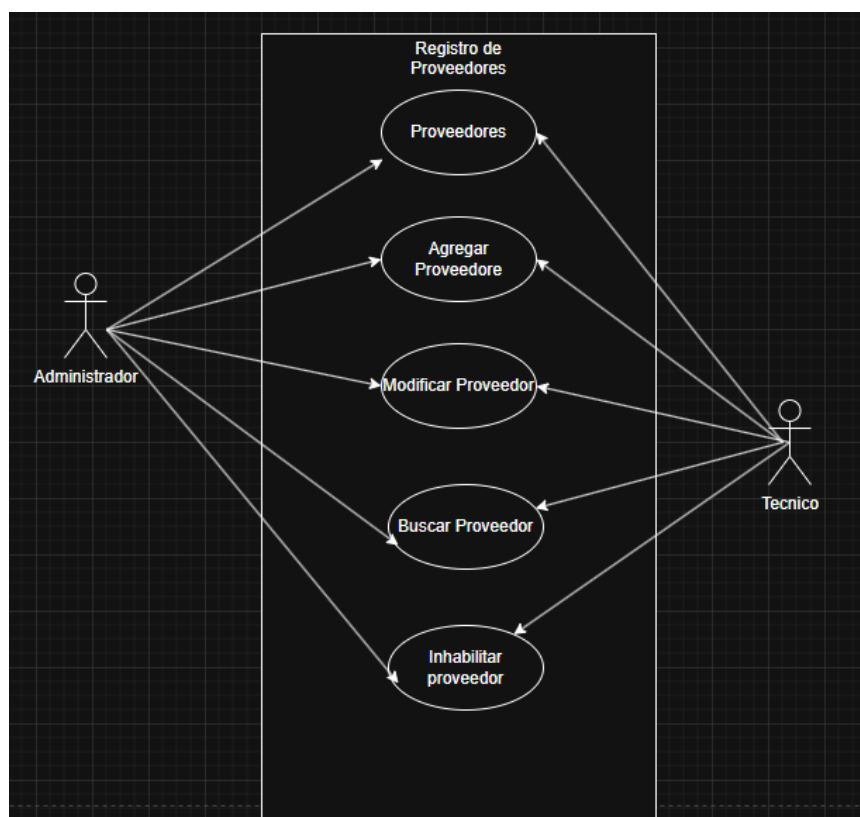
Tabla 35: Registro de Activos

Casos de uso	Registro de Activos
Autor(a)	Bryan Argüello Saborío

Actores	- Administrador - Sistema
Observaciones	- Lista los activos existentes - Agrega, modifica o inhabilita activos
Descripción	1. El administrador accede a la sección “Activos”. 2. Ingresa los datos del nuevo activo (nombre, código, estado, etc.). 3. Presiona “Guardar”. 4. El sistema valida los campos obligatorios. 5. Si es exitoso, se registra el nuevo activo en la base de datos. 2. Se alista un archivo CSV con los datos requeridos y se actualiza automáticamente el inventario.
Mensajes de error	- Si faltan campos obligatorios, el sistema muestra un mensaje. - Si el código ya existe, se notifica un conflicto.

Caso de uso 4: registro de proveedores

Figura 70: Diagrama de caso de uso de registro de proveedores



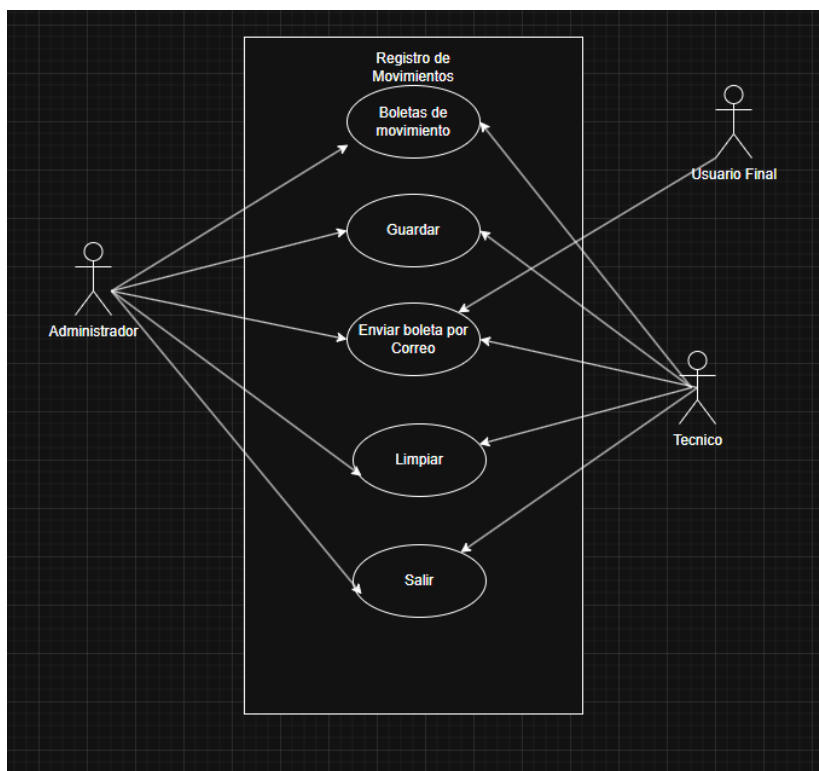
La figura 70 del diagrama de caso de uso pertenece al registro de proveedores del sistema de gestión de activos digitales de Instacredit.

Tabla 36: Registro Proveedores

Casos de uso	Registro de Proveedores
Autor(a)	Bryan Argüello Saborío
Actores	Administrador - Técnico
Observaciones	Lista los datos de proveedores Agrega, modifica o inhabilita proveedores
Descripción	1. El administrador ingresa a la sección “Proveedores”. 2. Completa los campos (nombre, contacto, etc.). 3. Presiona “Guardar”. 4. El sistema valida la información. 5. Se registra el proveedor en la base de datos.
Mensajes de error	Campos obligatorios incompletos - Proveedor ya registrado

Caso de uso 5: registro de boletas de movimiento

Figura 71: Diagrama de caso de uso de boletas de movimiento



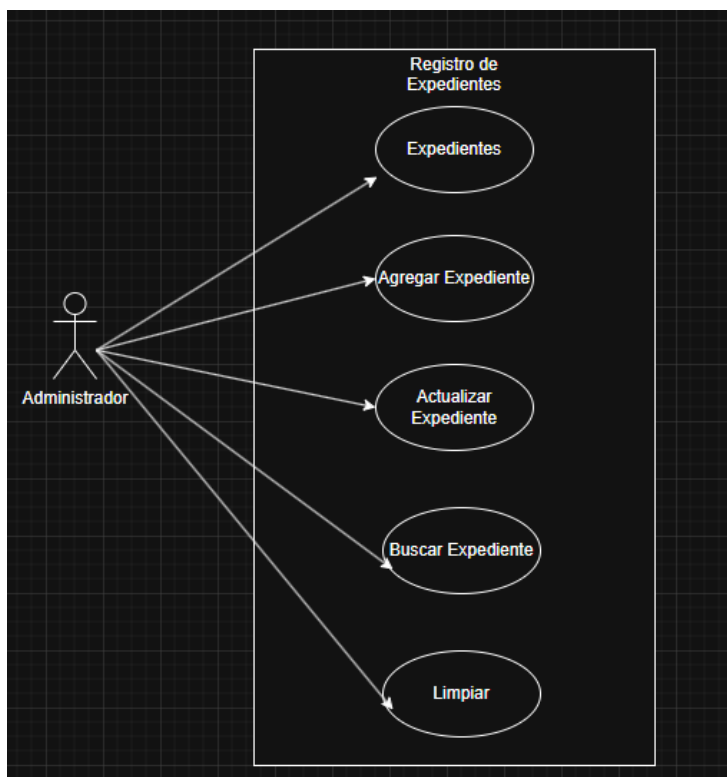
La figura 71 del diagrama de caso de uso pertenece al registro de movimientos del sistema de gestión de activos digitales de Instacredit.

Tabla 37: Registro de boletas de movimiento

Casos de uso	Boletas de Movimiento
Autor(a)	Bryan Argüello Saborío
Actores	Administrador - Técnico - Usuario final
Observaciones	Lista, crea, guarda envía por correo las boletas para documentar movimientos de activos
Descripción	1. El administrador o técnico accede a la sección “Boletas de Movimiento”. 2. Ingresar datos (tipo de movimiento, fecha, activo, responsable). 3. Presiona “Guardar”. 4. El sistema registra la boleta y actualiza el estado del activo si corresponde.
Mensajes de error	Datos incompletos - Activo no disponible o ya asignado

Caso de uso 6: registro de expedientes

Figura 72: Diagrama de caso de uso de expedientes



La figura 72 del diagrama de caso de uso pertenece al registro de expedientes del sistema de gestión de activos digitales de Instacredit.

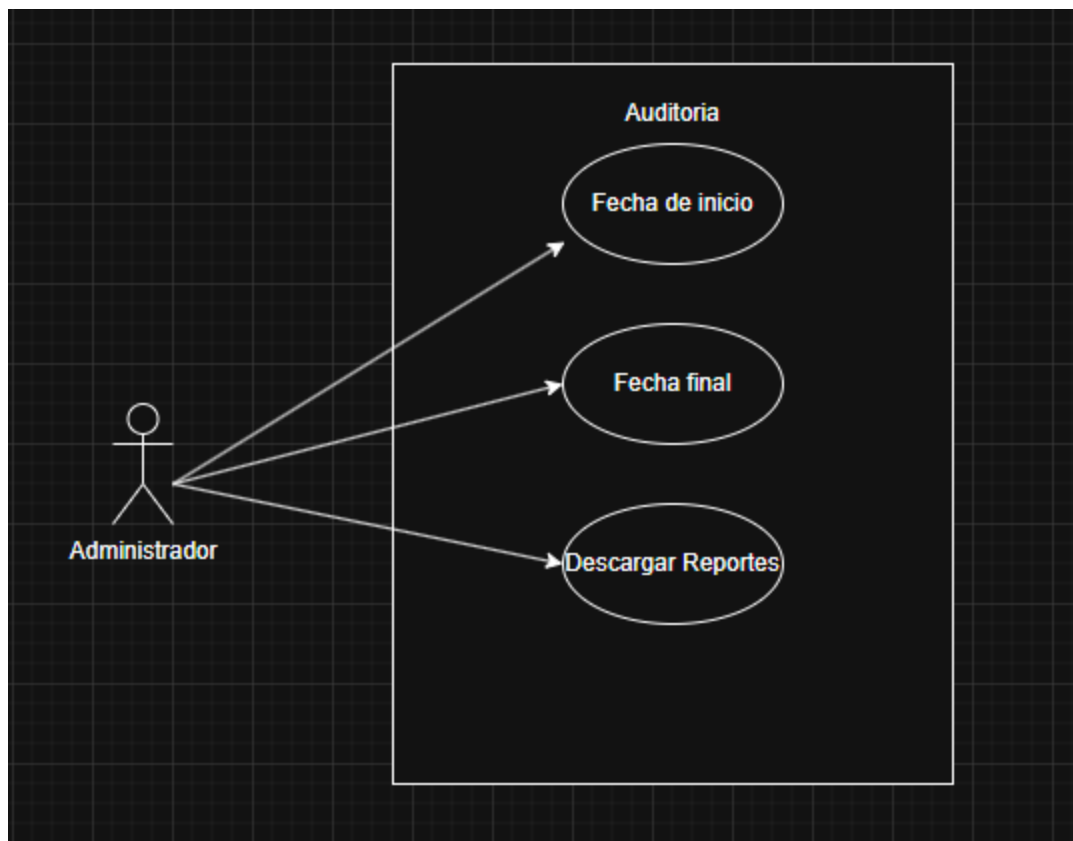
Tabla 38: Registro de expedientes

Casos de uso	Registro de Expedientes
Autor(a)	Bryan Argüello Saborío
Actores	Administrador - Usuario (con permisos)
Observaciones	Lista y documenta incidencias, amonestaciones o soporte técnico
Descripción	1. El actor con permisos ingresa a "Expedientes". 2. Completa la información requerida (descripción, fecha, activo involucrado). 3. Presiona "Guardar". 4. El sistema valida los campos y registra el expediente en la base de datos.

Mensajes de error	- Faltan campos obligatorios - Error al guardar
-------------------	---

Caso de uso 7: Auditoría

Figura 73: Diagrama de caso de uso de Auditoría



La figura 73 del diagrama de caso de uso pertenece al registro de auditoría del sistema de gestión de activos digitales de Instacredit.

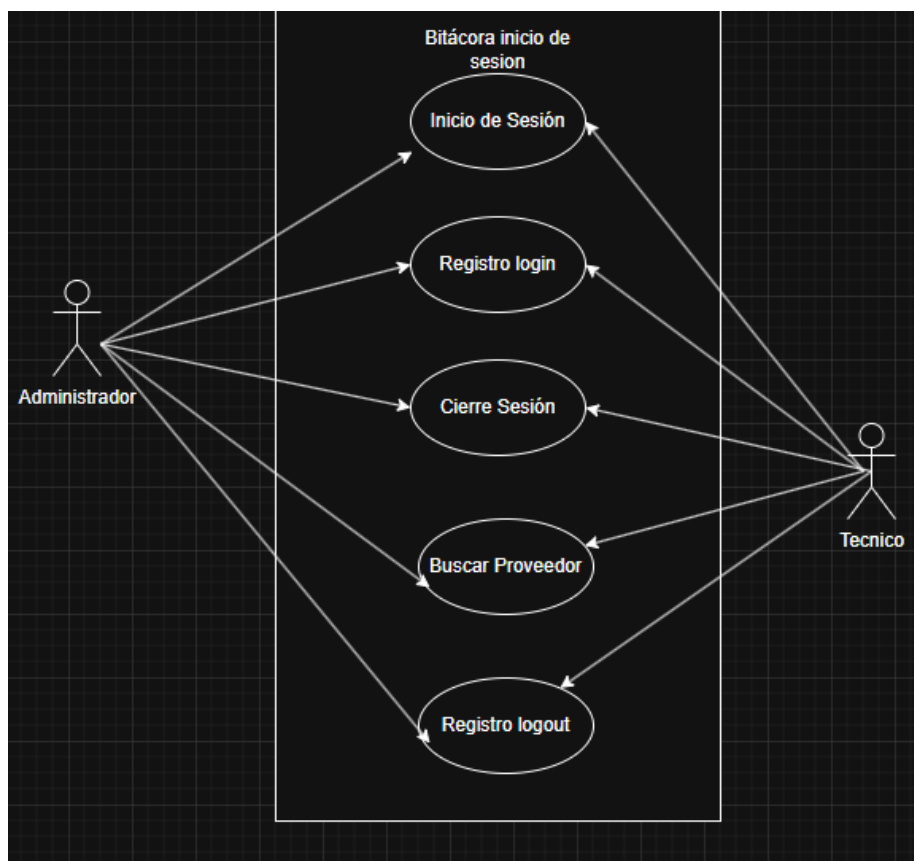
Tabla 39: Auditoría

Casos de uso	Auditoría
Autor(a)	Bryan Argüello Saborío

Actores	- Administrador - Sistema
Observaciones	- Visualiza y filtra registros de cambios realizados en el sistema
Descripción	1. El administrador accede a la sección “Auditoría”. 2. Puede buscar por fecha. 3. El sistema muestra la lista de eventos registrados. 4. El administrador puede exportar el reporte en pdf.
Mensajes de error	- N/A (generalmente no hay mensajes de error específicos, salvo fallas de conexión)

Caso de uso 8: bitácora inicio de sesión

Figura 74: Diagrama de caso de uso bitácora inicio de sesión



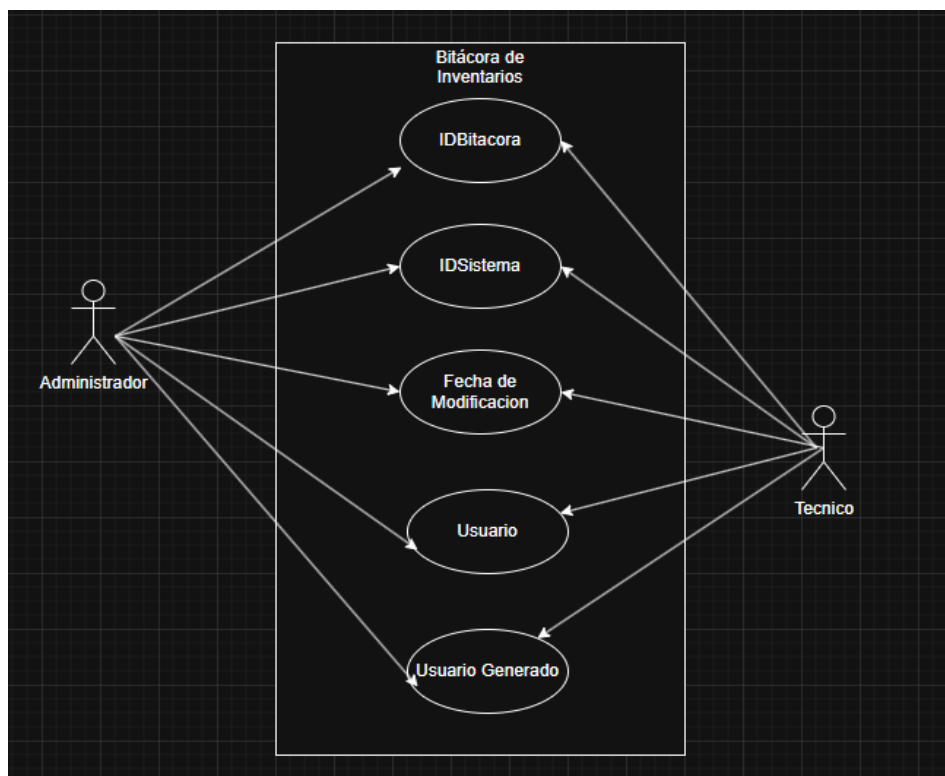
La figura 74 del diagrama de caso de uso pertenece al registro de bitácora de inicio de sesión del sistema de gestión de activos digitales de Instacredit.

Tabla 40: Bitácora de sesiones

Casos de uso	Control de Bitácora de Sesiones
Autor(a)	Bryan Argüello Saborío
Actores	- Administrador - Sistema
Observaciones	- Muestra quién ingresó, a qué hora y cuándo cerró sesión
Descripción	1. El usuario (Administrador/Usuario) inicia sesión (ver Caso de Uso: Login). 2. El sistema registra la fecha y hora de inicio. 3. Al cerrar sesión, se registra la fecha y hora de salida. 4. El administrador puede consultar esta bitácora para auditar accesos.
Mensajes de error	Error de conexión con la base de datos.

Caso de uso 9: bitácora de inventarios

Figura 75: Bitácora de inventarios



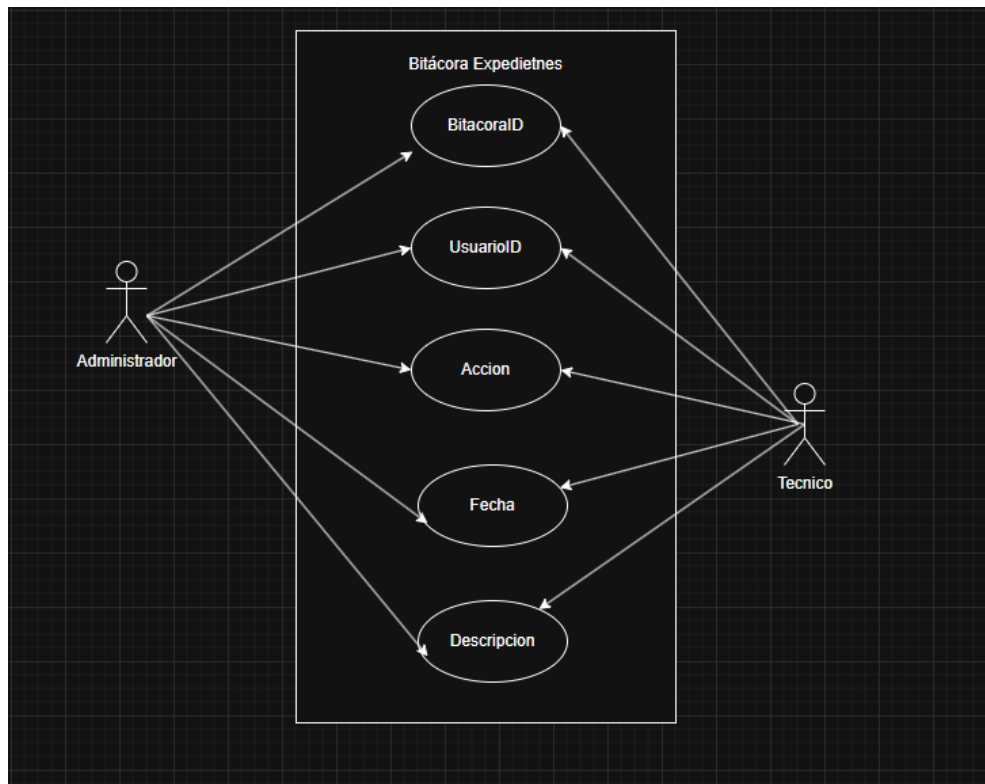
La figura 75 del diagrama de caso de uso pertenece al registro de bitácora de inventarios del sistema de gestión de activos digitales de Instacredit.

Tabla 41: Bitácora de Inventarios

Casos de uso	Bitácora de Inventarios
Autor(a)	Bryan Argüello Saborío
Actores	- Administrador - Usuario (con permisos) - Sistema
Observaciones	- El sistema registra cada modificación o movimiento en el inventario - Permite filtrar registros por fecha, usuario o tipo de cambio
Descripción	1. El actor (Administrador/Usuario con permisos) accede a la sección “Bitácora de Inventarios”. 2. El sistema muestra la lista de registros obtenida de la tabla bitácora inventario (ID Bitácora, ID Sistemas, Fecha Modificación, Usuario, Usuario Generado, etc.). 3. El actor puede filtrar por rango de fechas, tipo de acción (agregar, actualizar, inhabilitar, etc.) o responsable. 4. El sistema despliega los resultados y permite exportar o consultar detalles adicionales (Descripción, Comentarios).
Mensajes de error	Error de conexión con la base de datos.

Caso de uso 10: bitácora de expedientes

Figura 76: Bitácora de expedientes



La figura 76 del diagrama de caso de uso pertenece al registro de bitácora de expedientes del sistema de gestión de activos digitales de Instacredit.

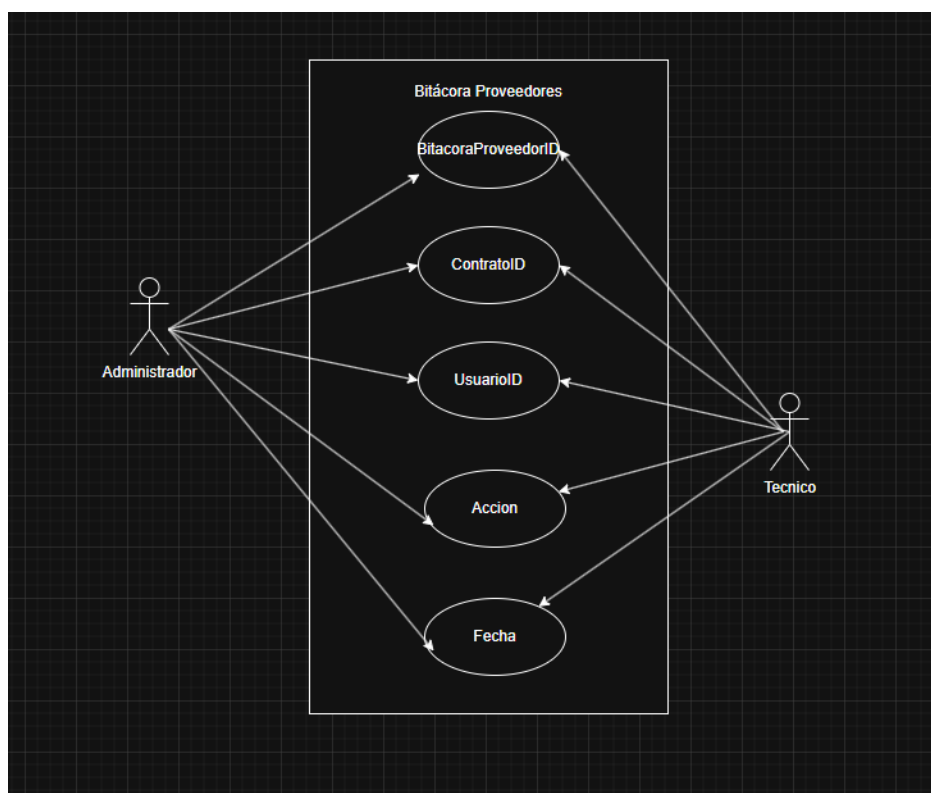
Tabla 42: Bitácora de Expedientes

Casos de uso	Bitácora de Expedientes
Autor(a)	Bryan Argüello Saborío
Actores	- Administrador - Usuario (con permisos) - Sistema
Observaciones	- El sistema registra y muestra los cambios realizados sobre los expedientes (contratos) - Permite filtrar registros por usuario, fecha y tipo de acción (Acción, Fecha, Descripción, etc.)

Descripción	1. El actor accede a la sección “Bitácora de Expedientes”. 2. El sistema consulta la tabla Bitácora Contratos (Bitácora Contrato ID, Contrato ID, Usuario ID, Acción, Fecha, Descripción). 3. El actor filtra por fecha, acción o usuario responsable. 4. El sistema muestra los registros correspondientes y, si se requiere, detalla la descripción de cada cambio.
Mensajes de error	- Error de conexión con la base de datos. - Falta de permisos para ver la bitácora.

Caso de uso 11: bitácora de proveedores

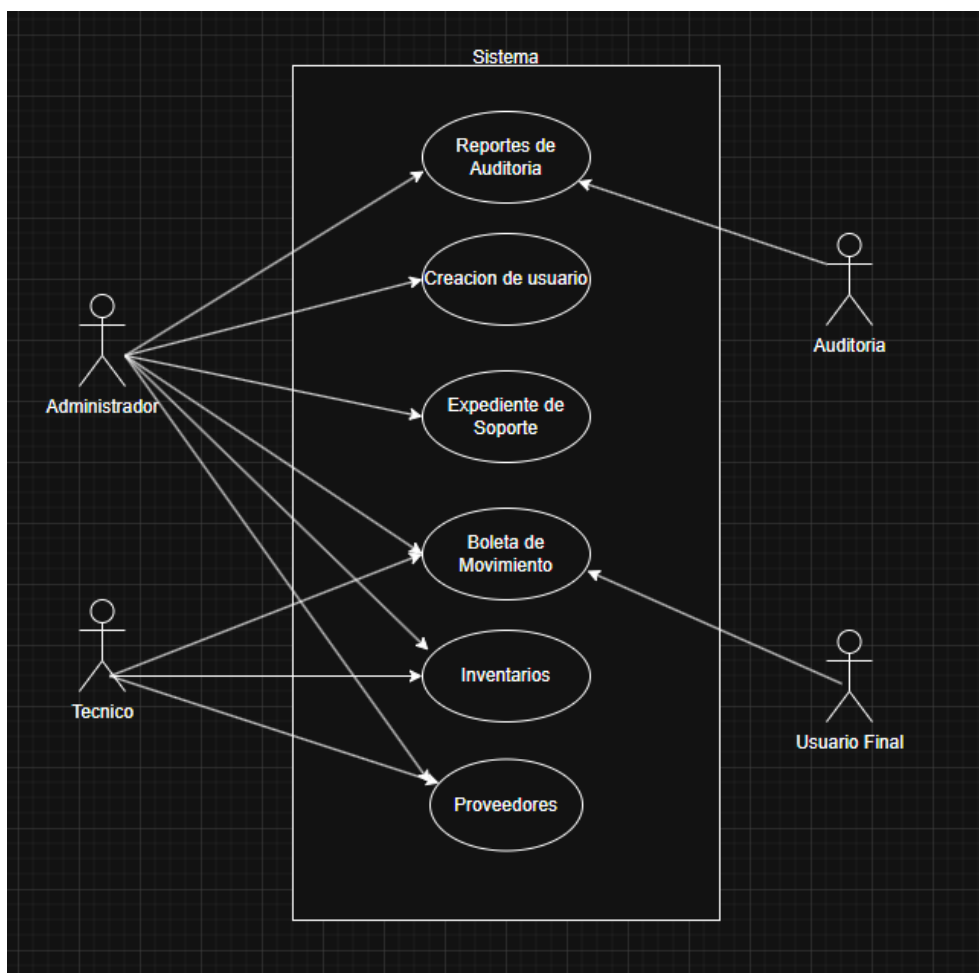
Figura 77: Bitácora de proveedores



La figura 77 del diagrama de caso de uso pertenece al registro de bitácora de proveedores del sistema de gestión de activos digitales de Instacredit.

Caso de uso 11: panel de control

Figura 78: Bitácora de proveedores



La figura 78 del diagrama de caso de uso pertenece al registro de panel de control del sistema de gestión de activos digitales de Instacredit.

Tabla 43: Panel de Control

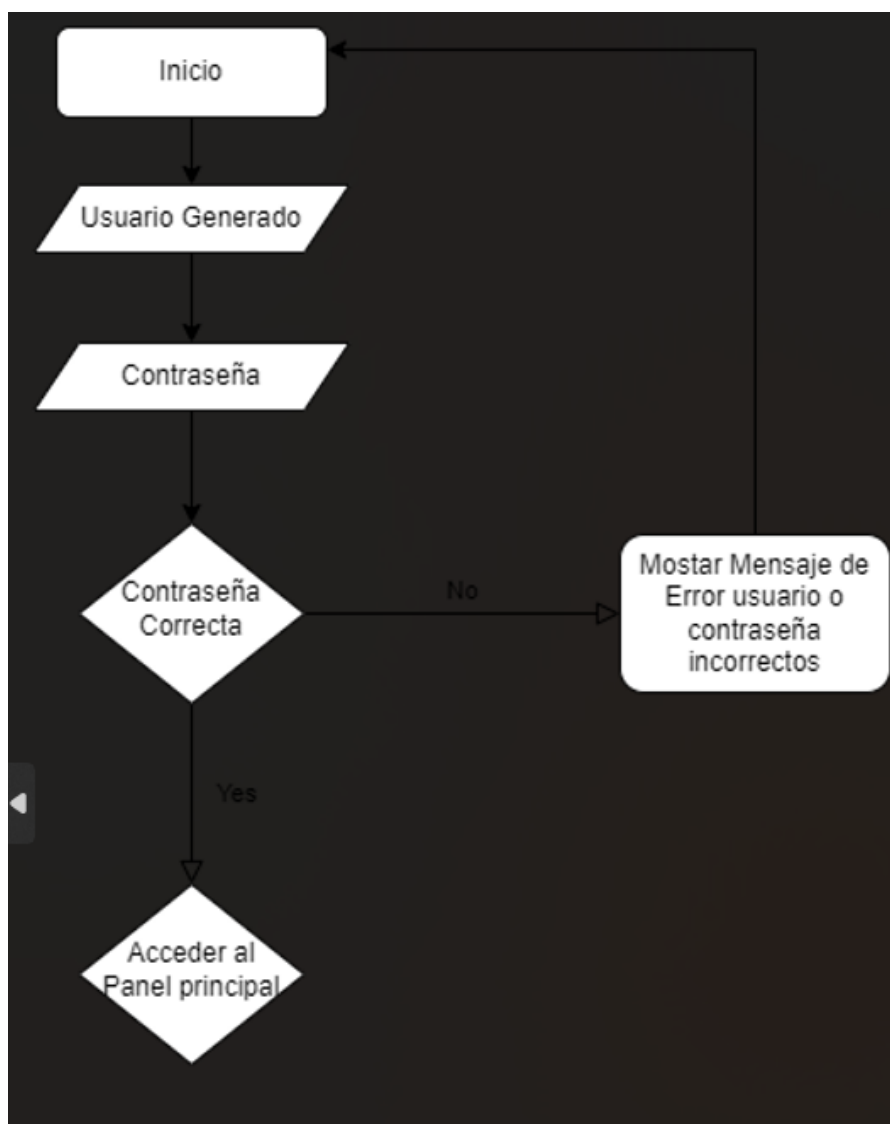
Casos de uso	Panel de control
Autor(a)	Bryan Argüello Saborío
Actores	- Administrador - Usuario (con permisos) - Sistema
Observaciones	Muestra todas las pestañas que se tienen disponibles en el sistema (Agregar Usuarios, Auditoría, Expedientes, Movimientos, Proveedores e Inventarios)

Descripción	1. El actor accede al sistema tras iniciar sesión. 2. El sistema presenta un panel o página principal con las pestañas disponibles según su rol .
Mensajes de error	Si no cuenta con los permisos correctos, salta alerta de que no tiene permisos de administrador para acceder a ciertas áreas.

Diagramas de flujo

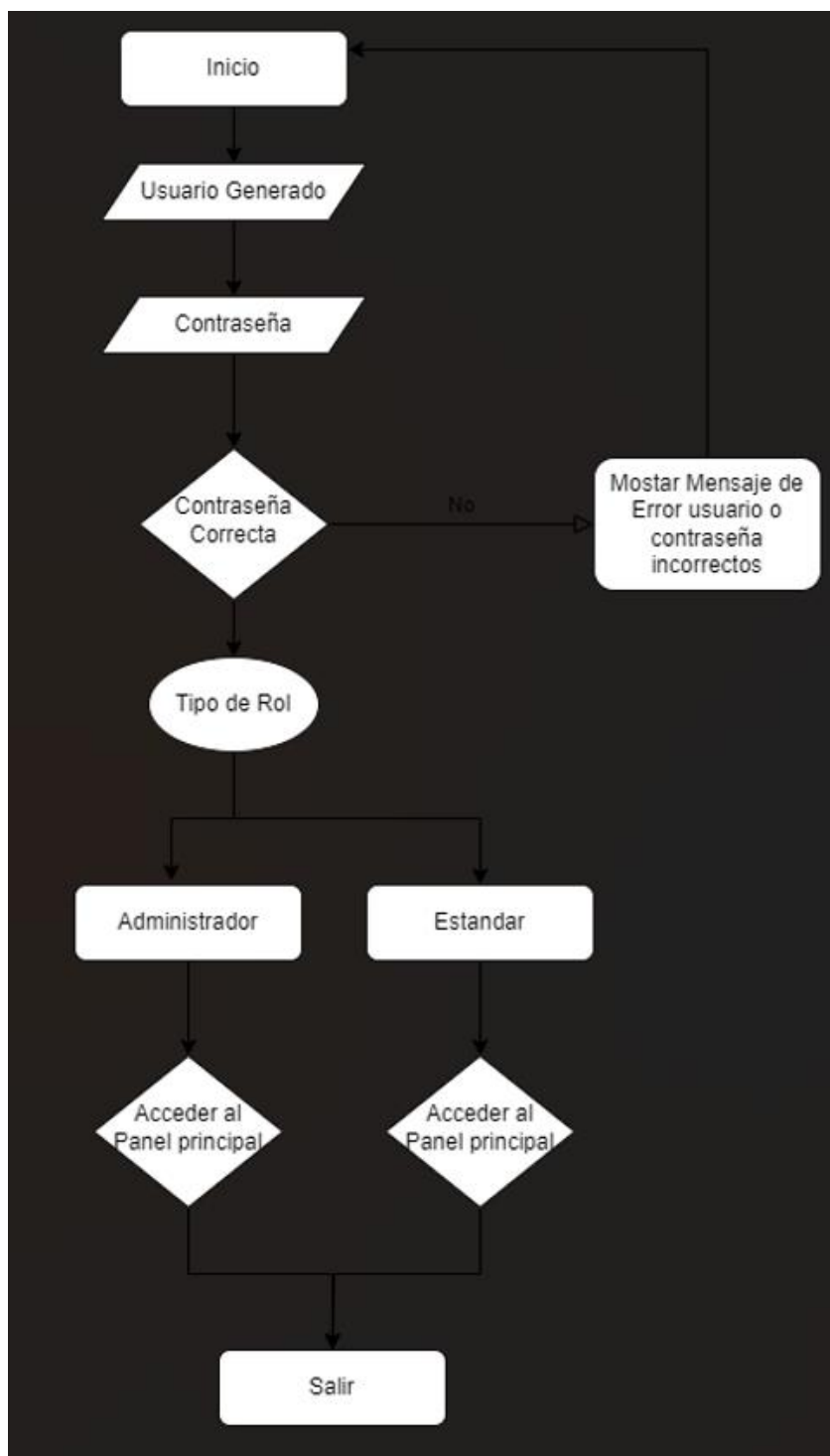
Login

Figura 79: Diagrama de flujo Login



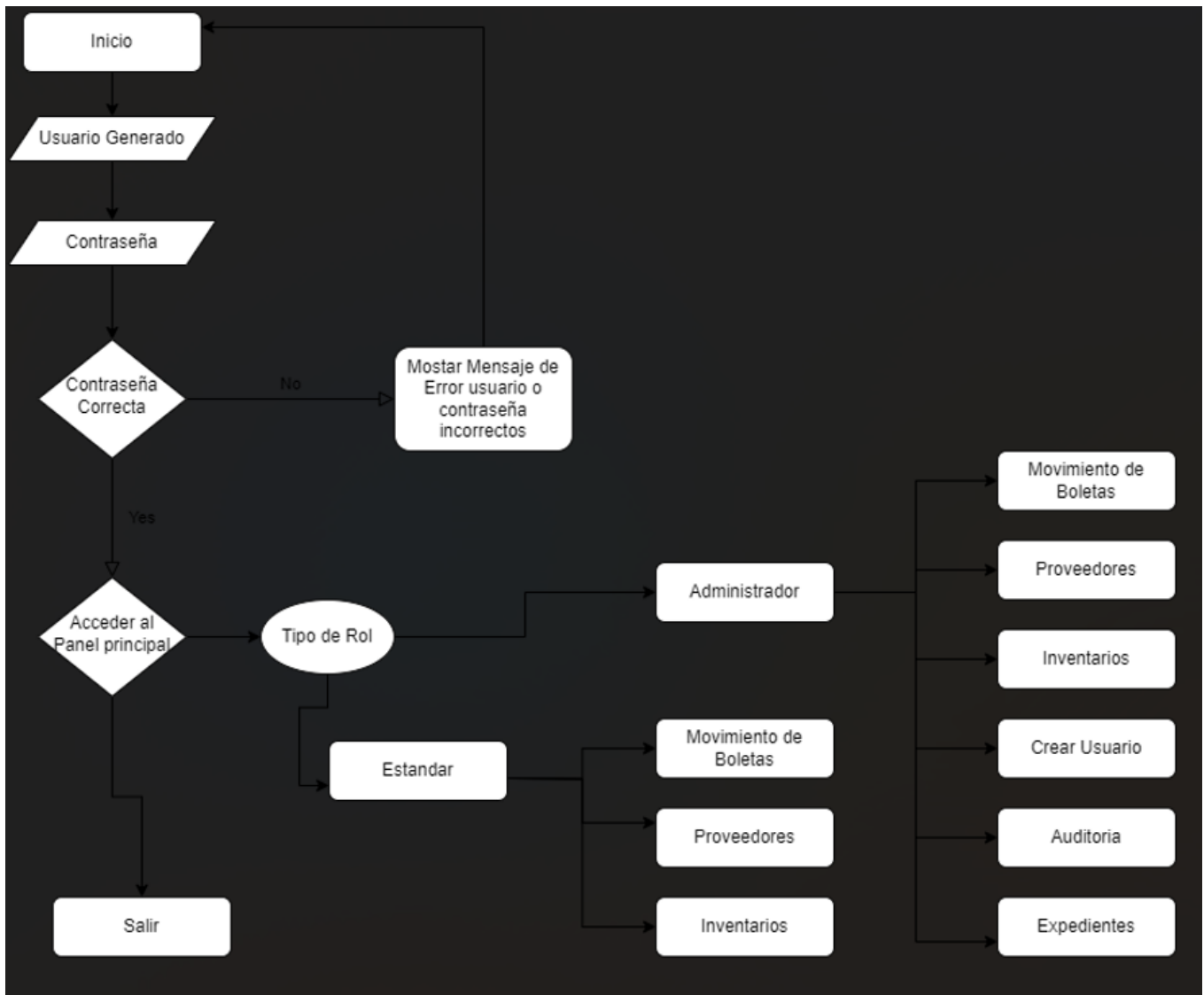
Panel de control

Figura 80: Diagrama de flujo Panel de control



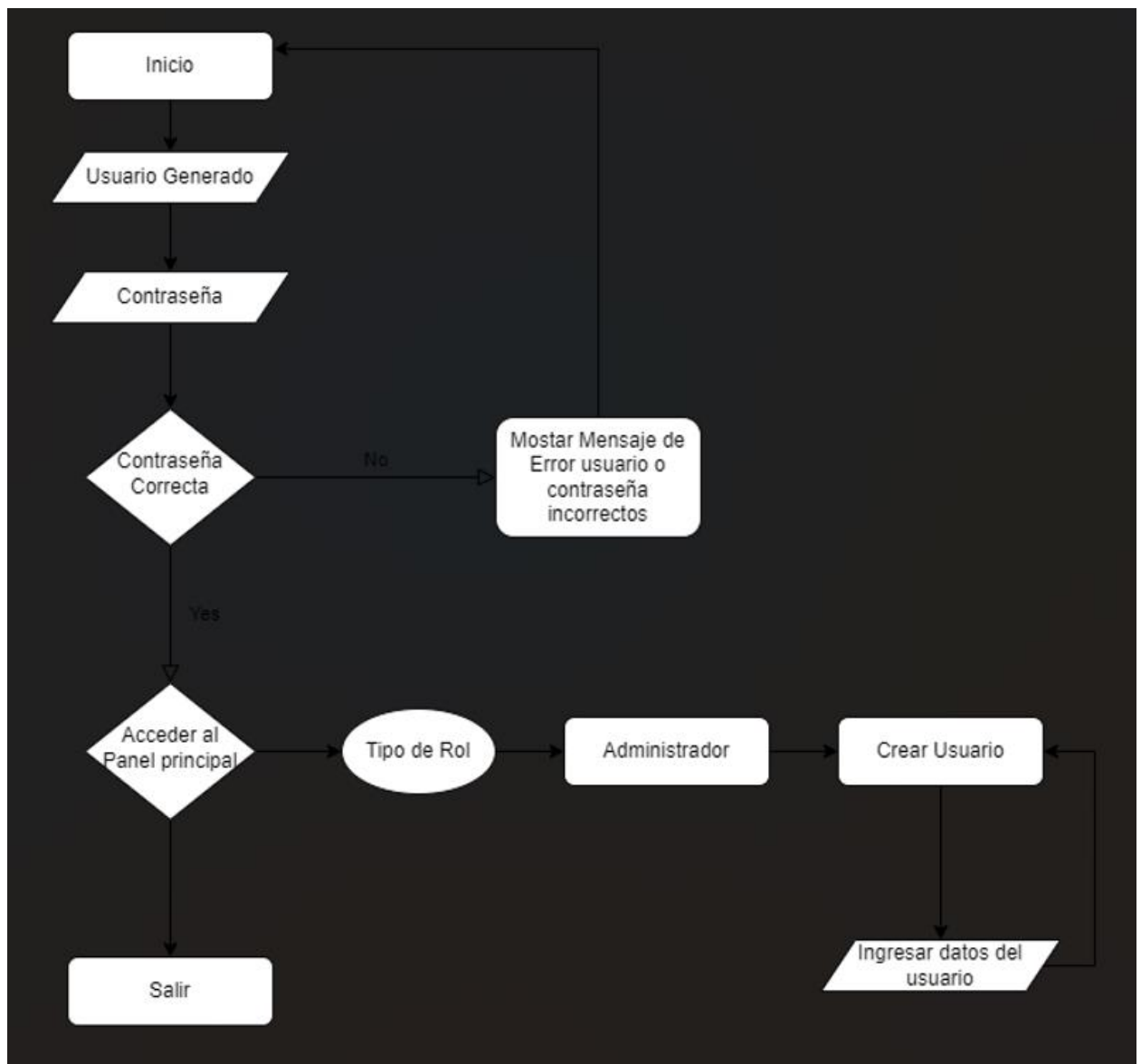
Tipo de rol

Figura 81: Diagrama de flujo Tipo de rol



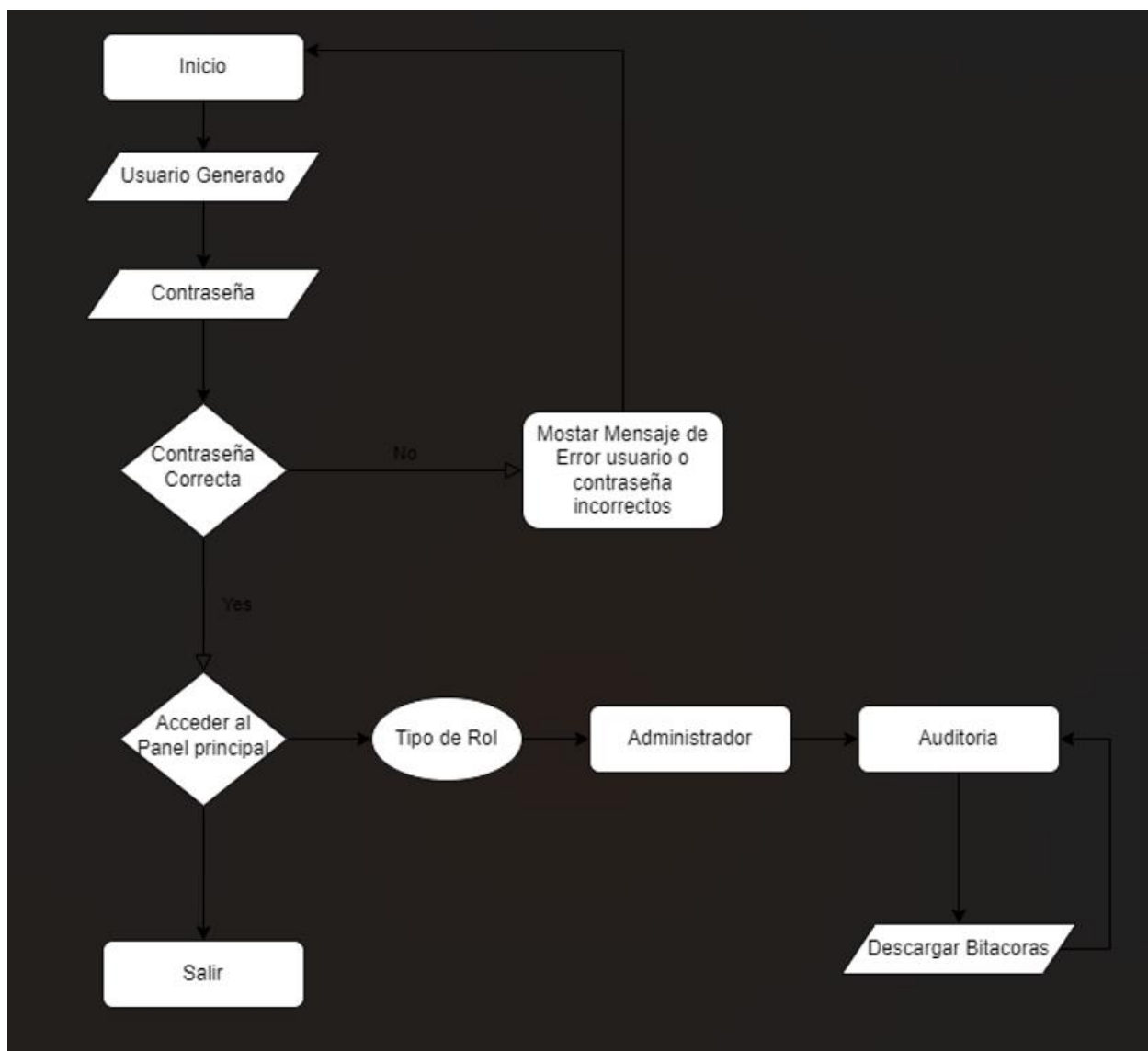
Registro usuario

Figura 82: Diagrama de flujo Registro usuario



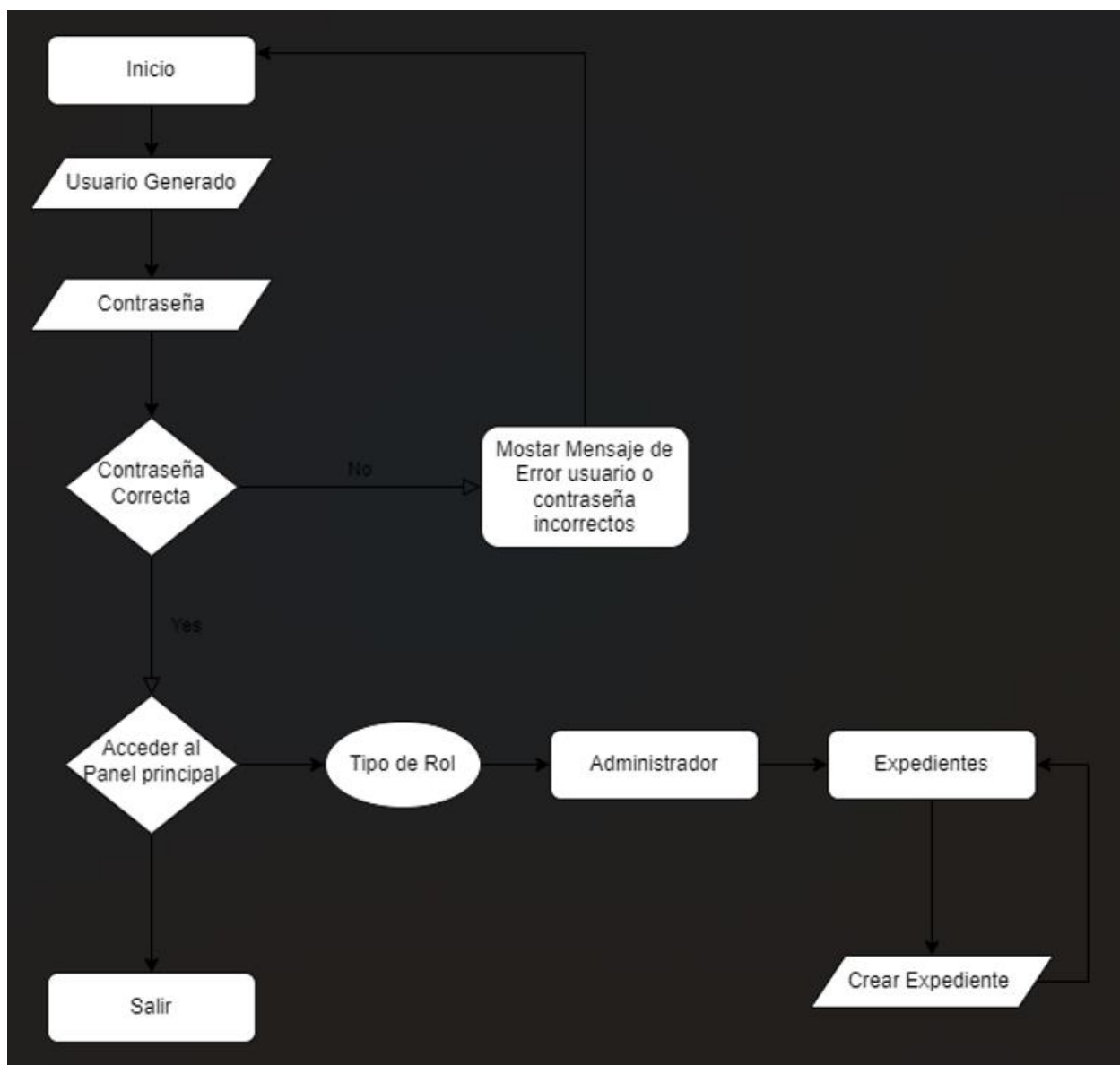
Reportes de Auditoría

Figura 83: Diagrama de flujo Reportes de Auditoría



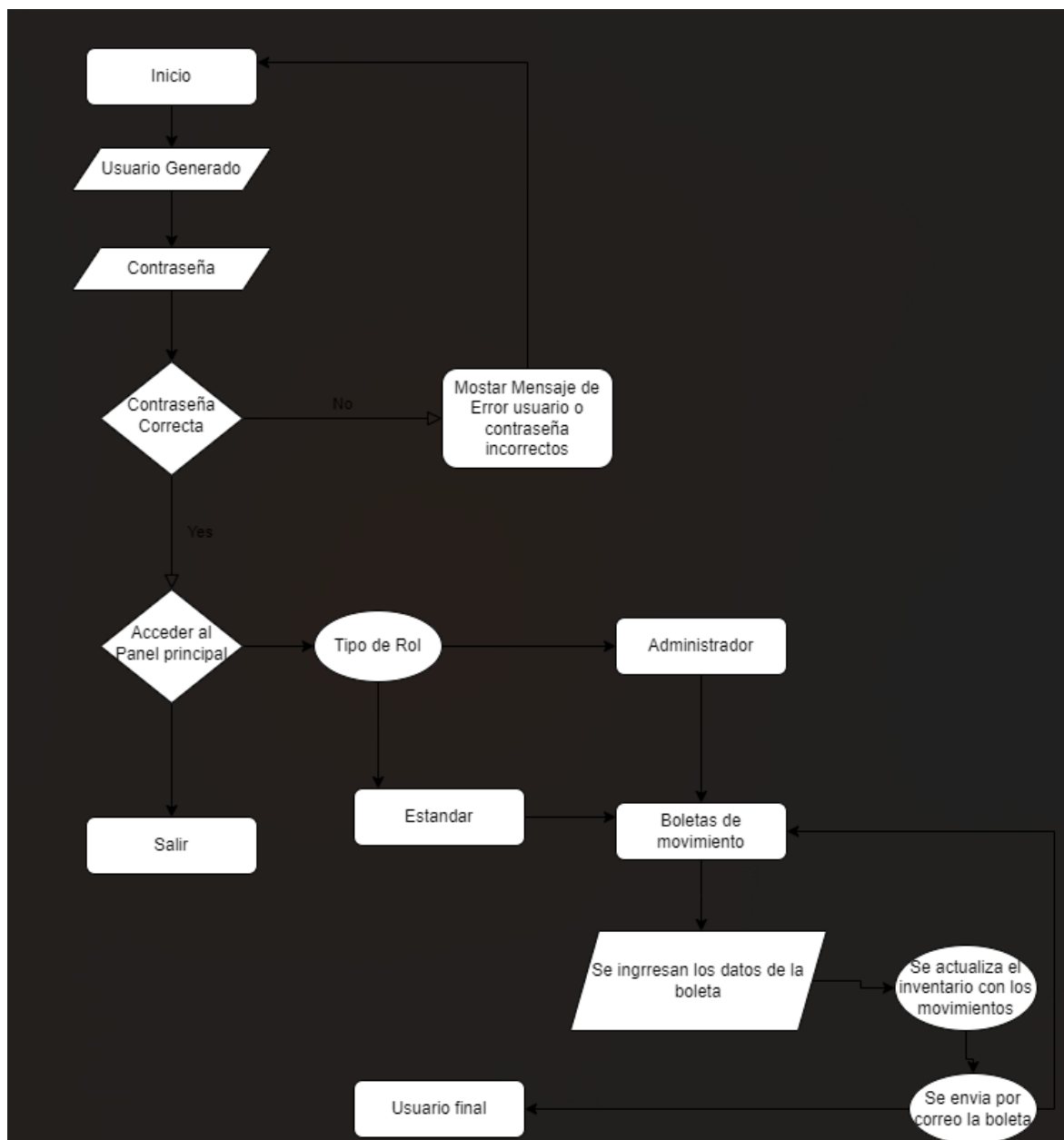
Expedientes de soporte

Figura 84: Diagrama de flujo Registro expedientes



Movimiento de boletas

Figura 85: Diagrama de flujo Registro movimiento de boletas



Diagramas de secuencia

Figura 87: Diagrama de secuencia Login

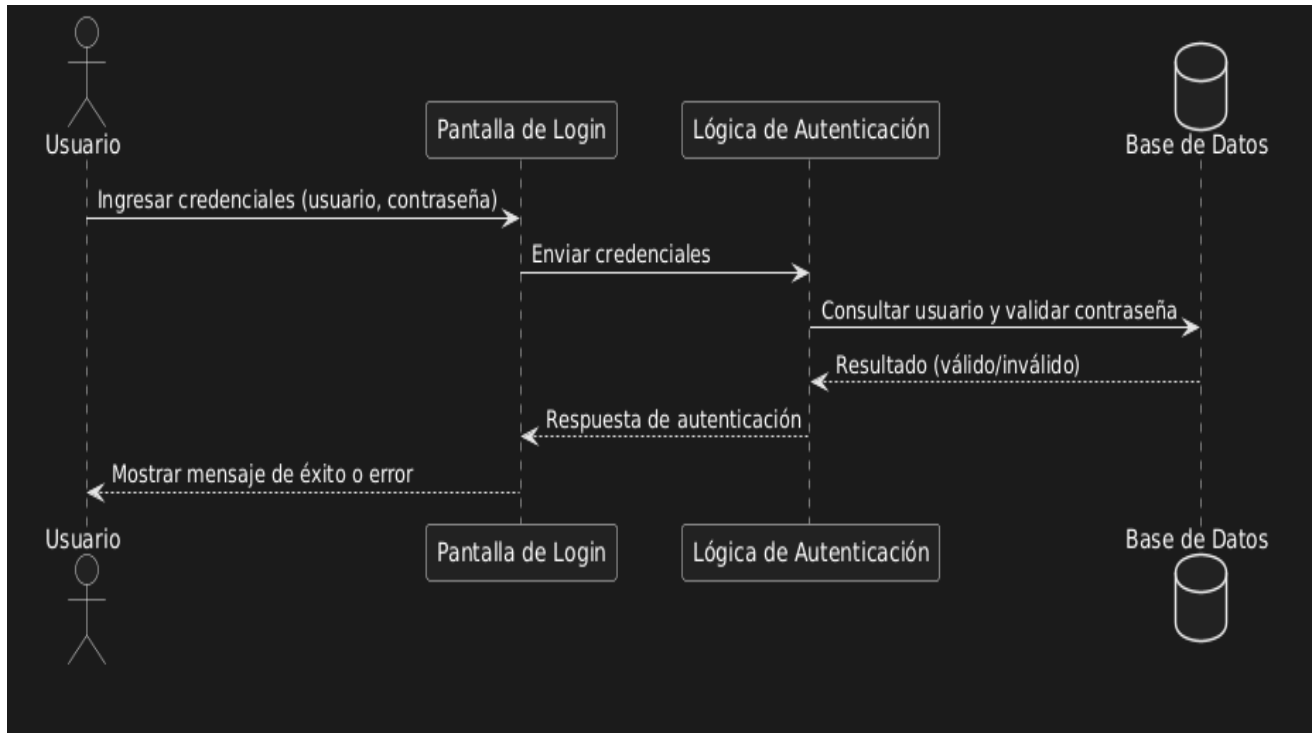


Figura 88: Diagrama de secuencia primer inicio de sesión

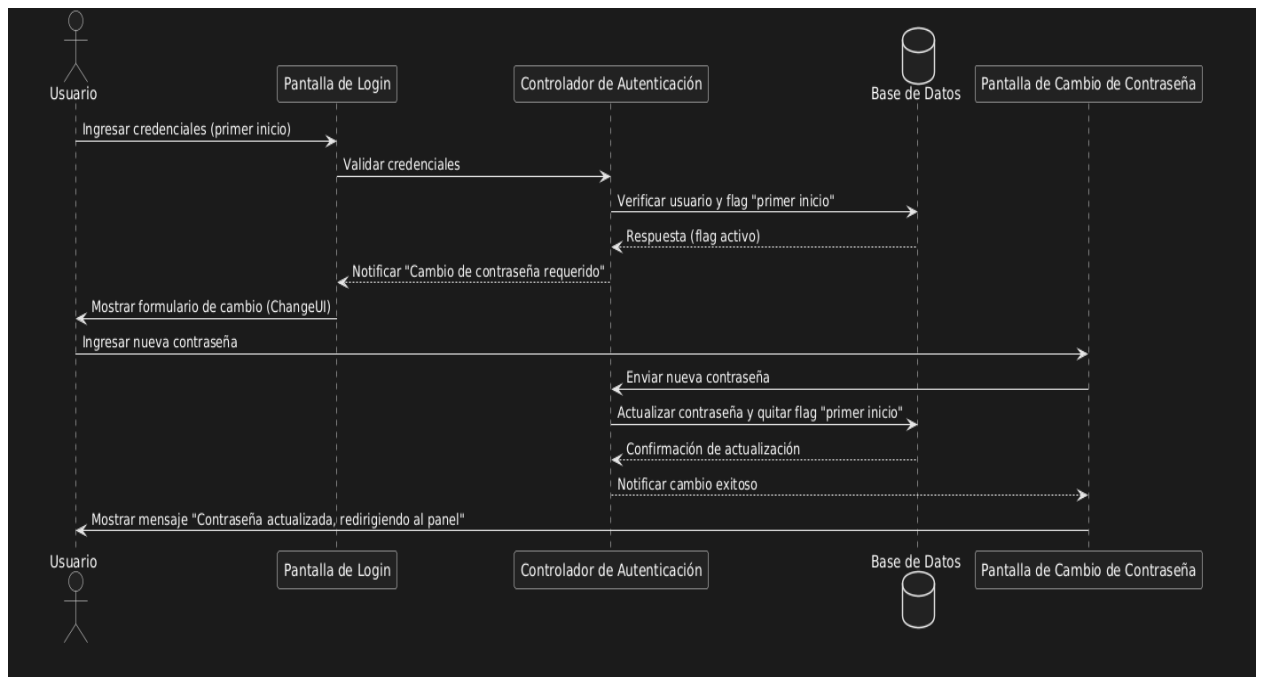


Figura 89: Diagrama de secuencia de olvido su contraseña

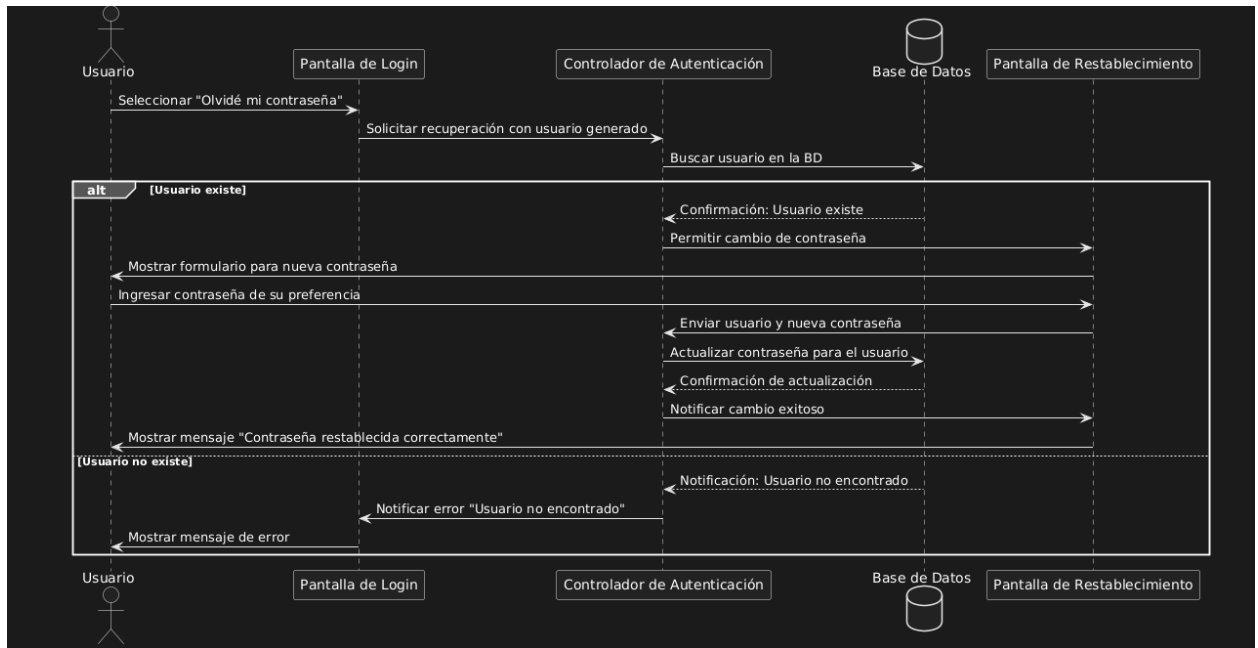


Figura 90: Diagrama de secuencia Panel de control

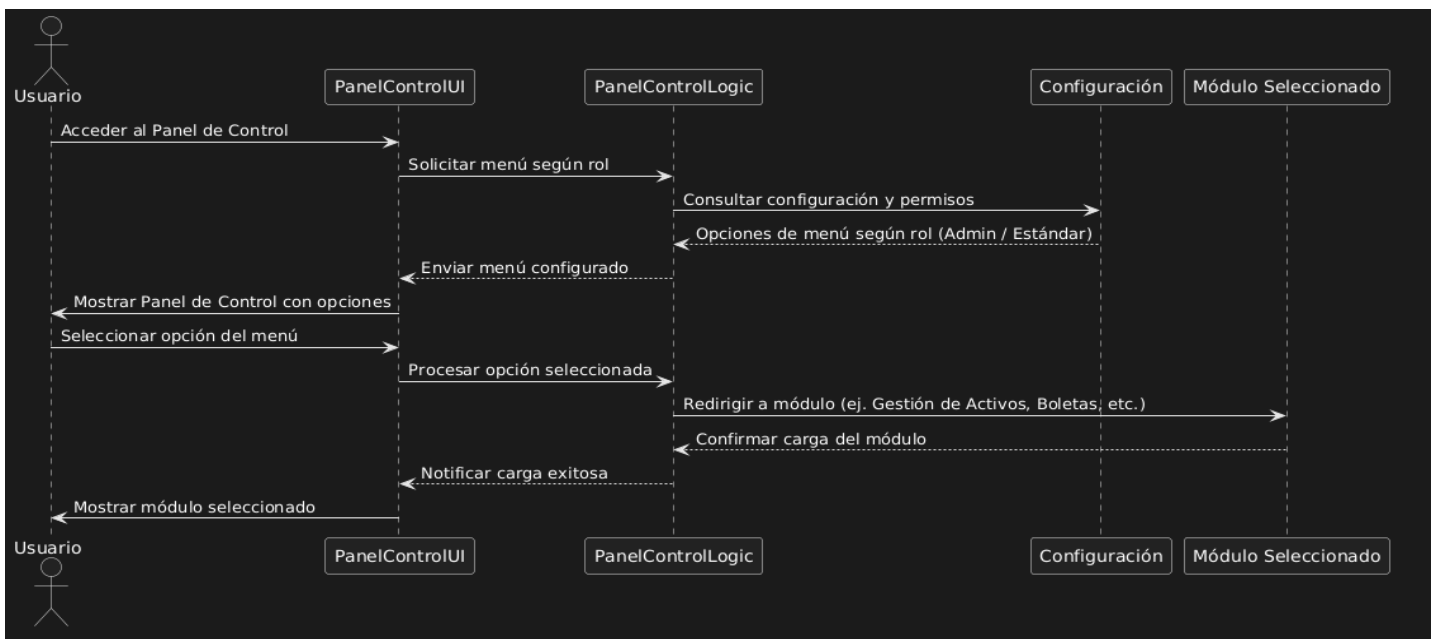


Figura 91: Diagrama de secuencia Agregar usuario

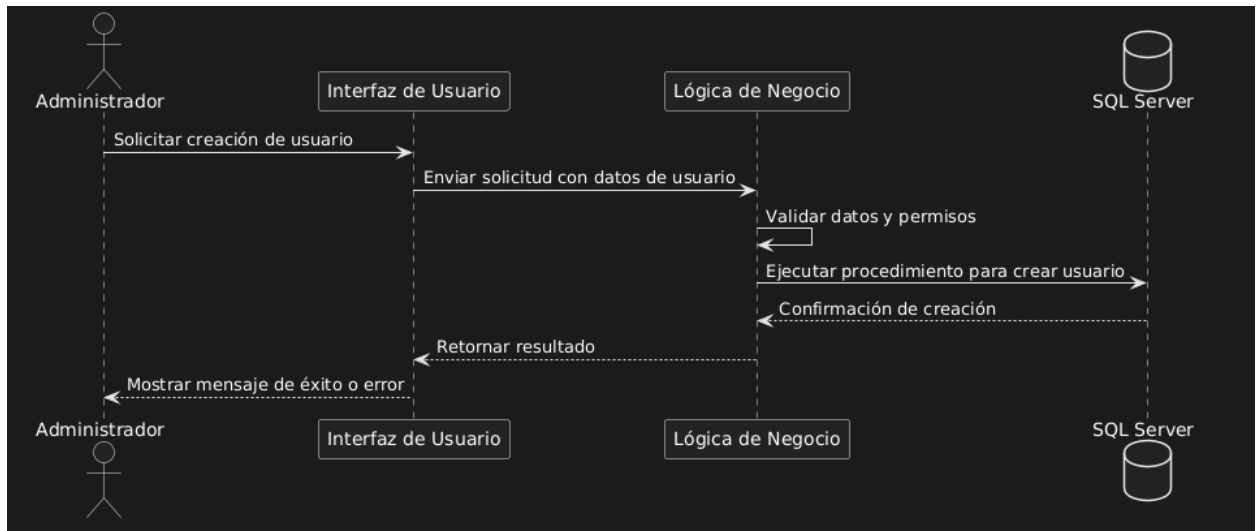


Figura 92: Diagrama de secuencia Auditoría

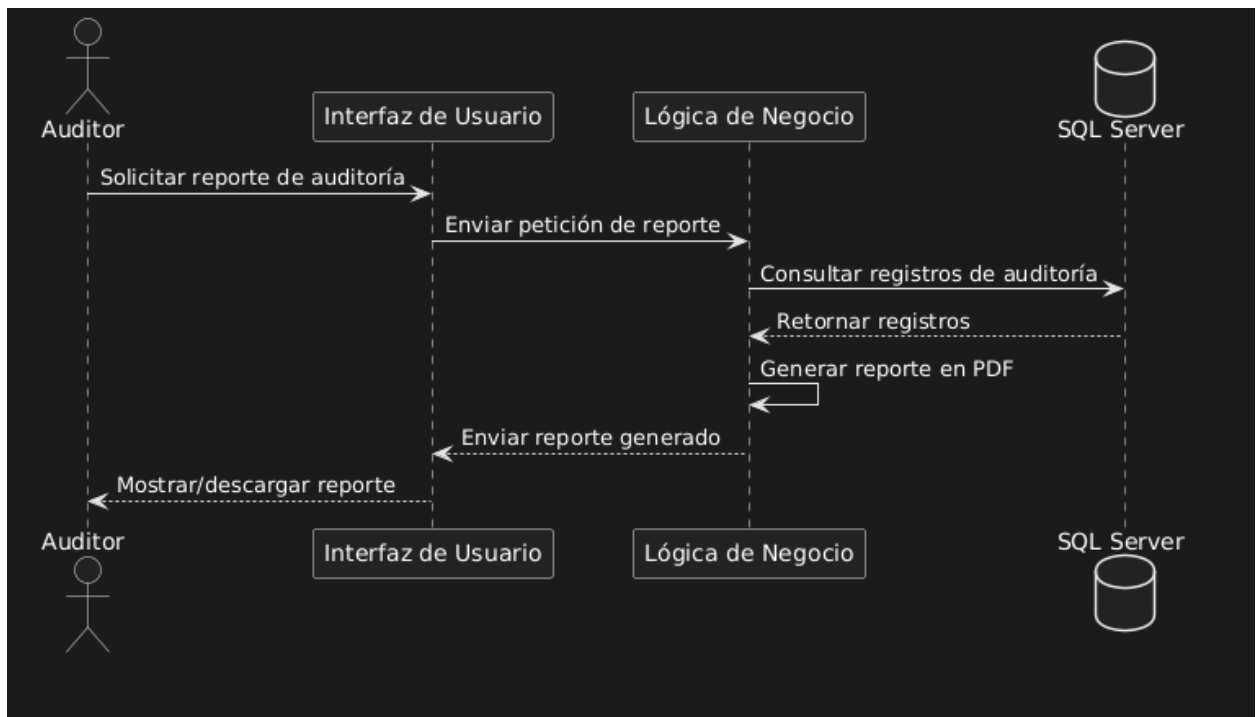


Figura 93: Diagrama de secuencia Expedientes

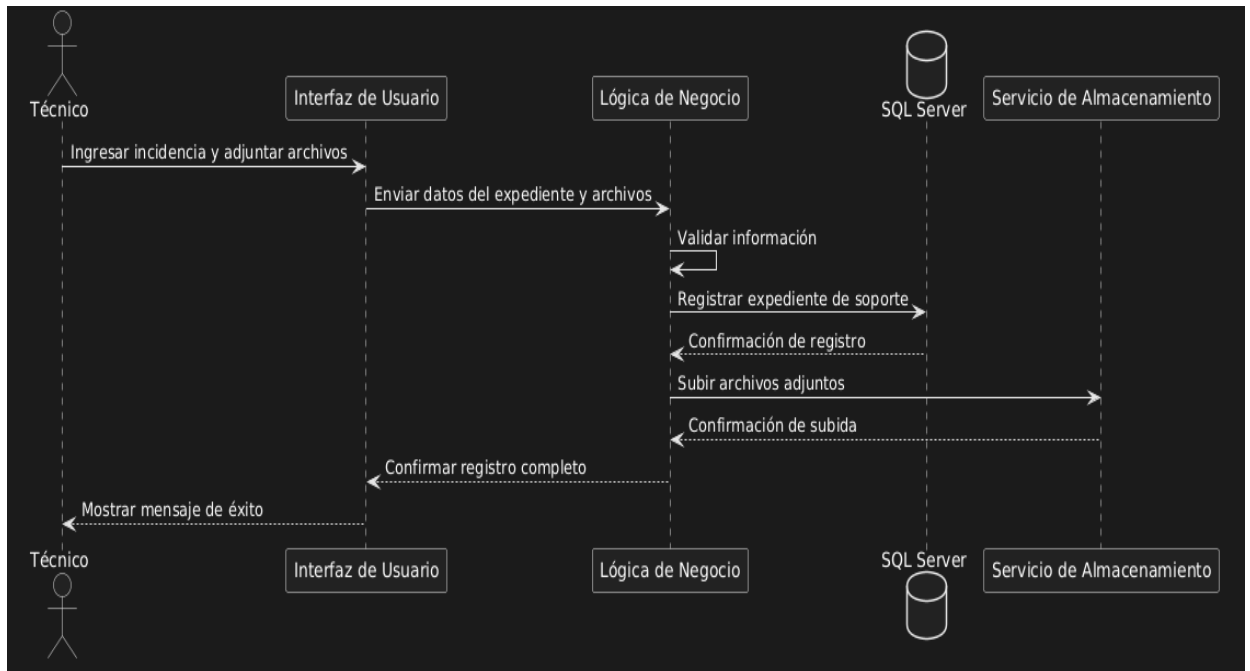


Figura 94: Diagrama de secuencia Movimiento de boletas

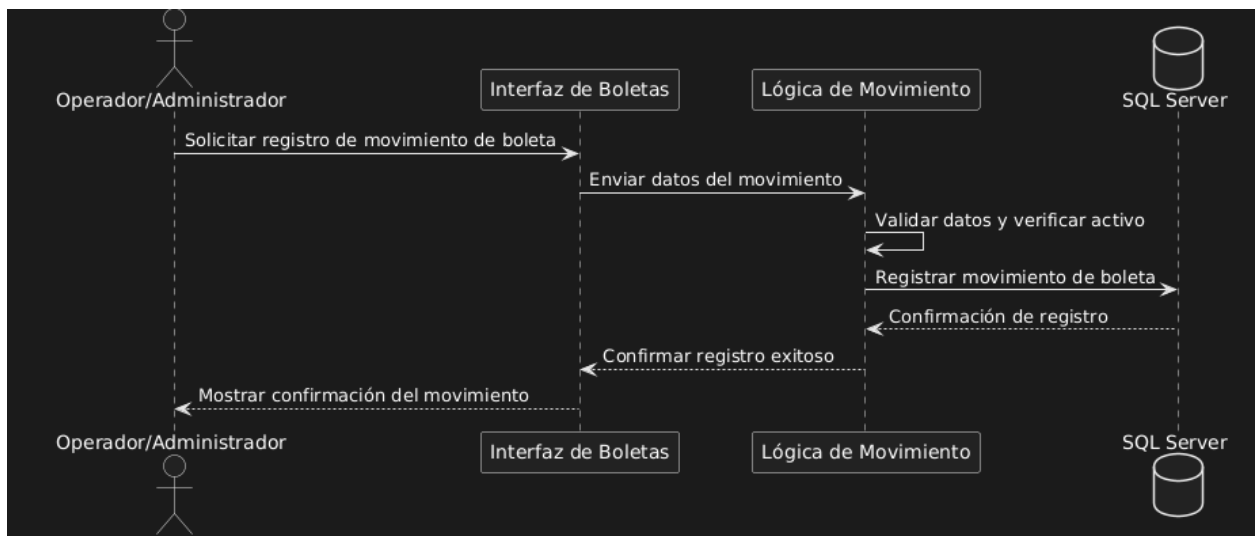


Figura 95: Diagrama de secuencia de proveedores

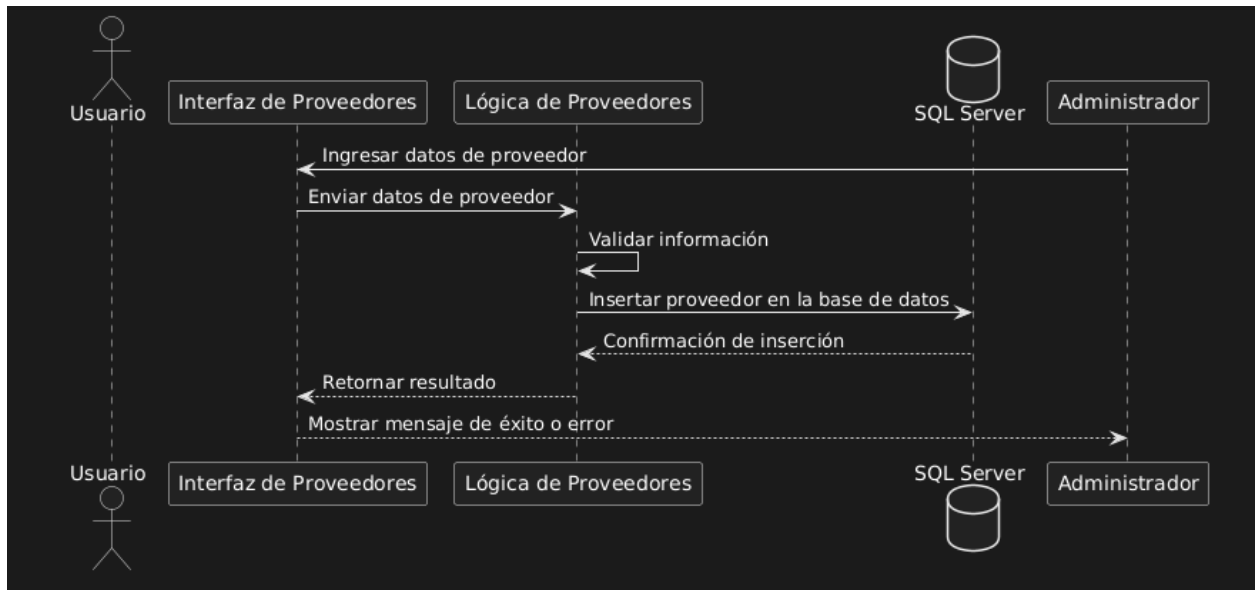


Figura 96: Diagrama de secuencia para agregar nuevo proveedor

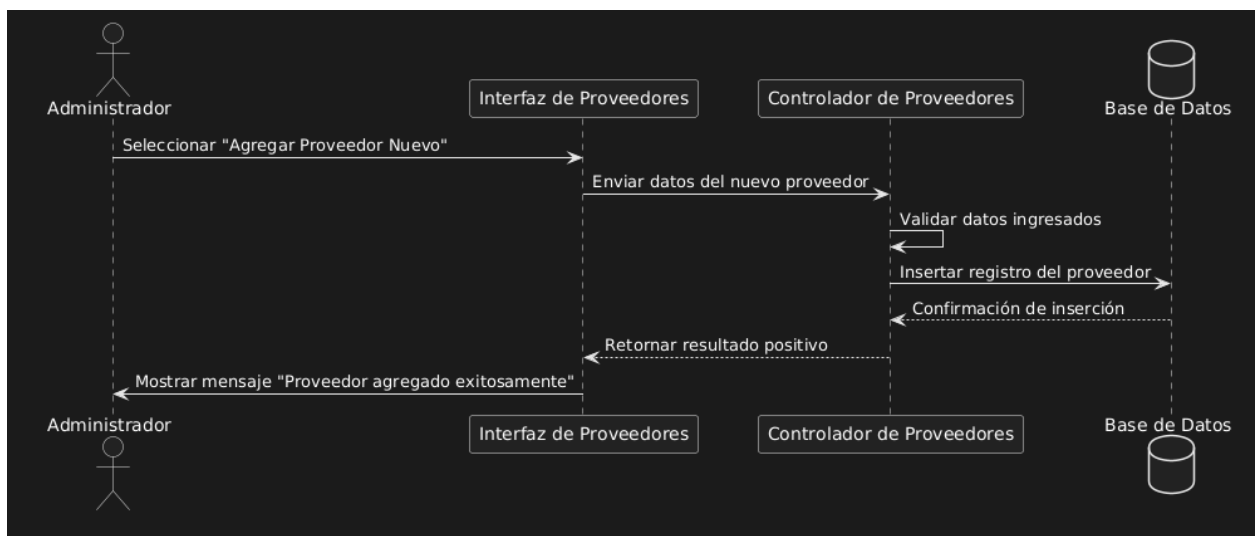
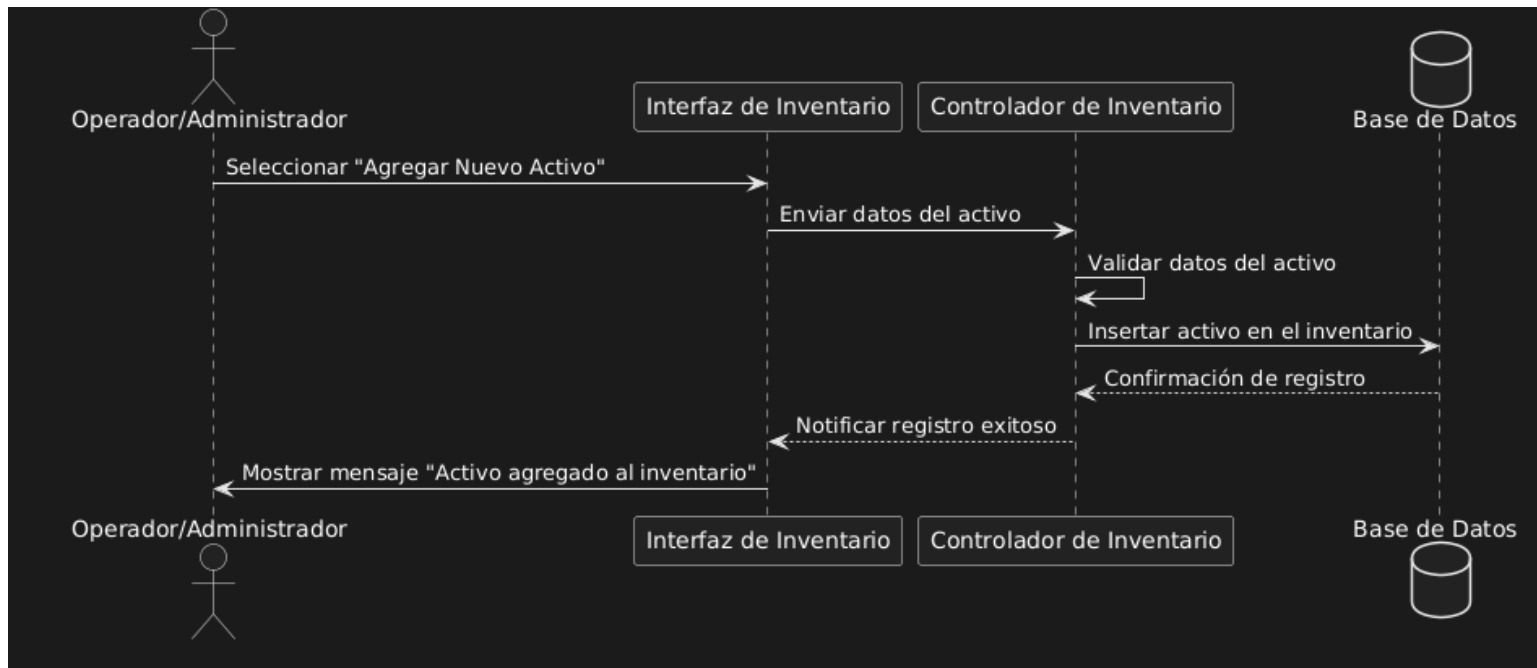


Figura 97: Diagrama de secuencia de inventarió



Modelo entidad-relación

Figura 98: Modelo relacional

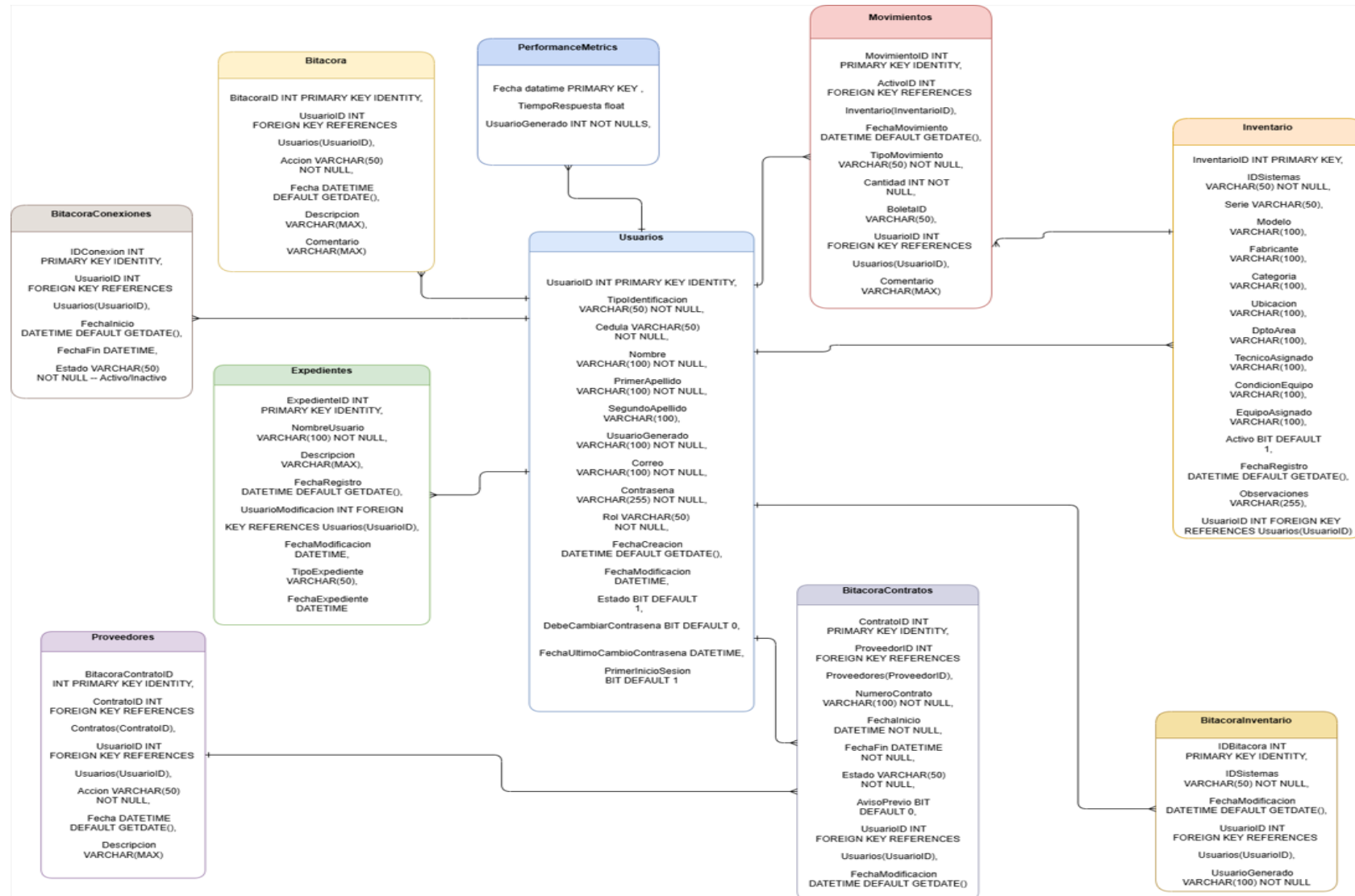


Figura 99:Modelo relacional

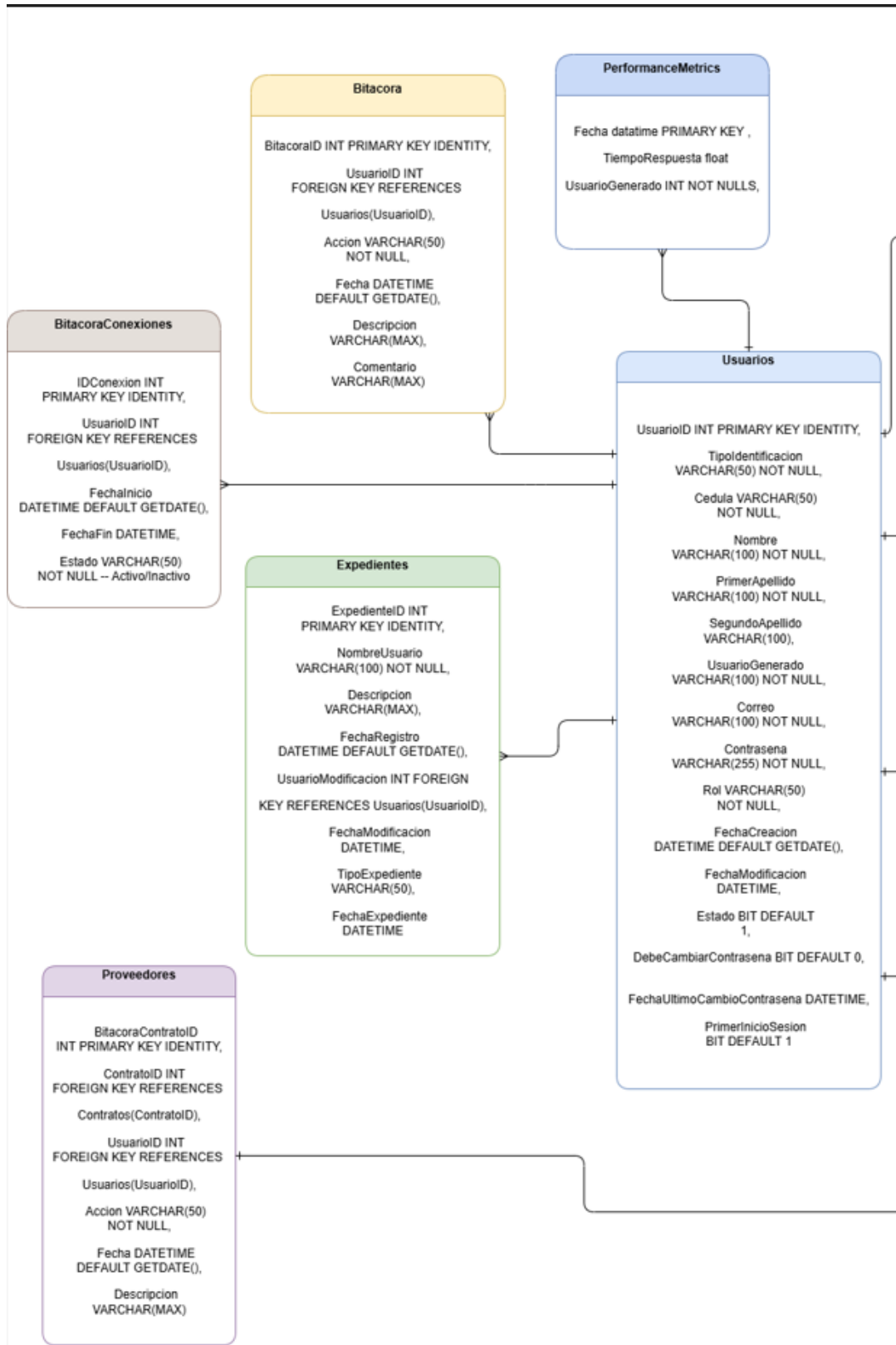
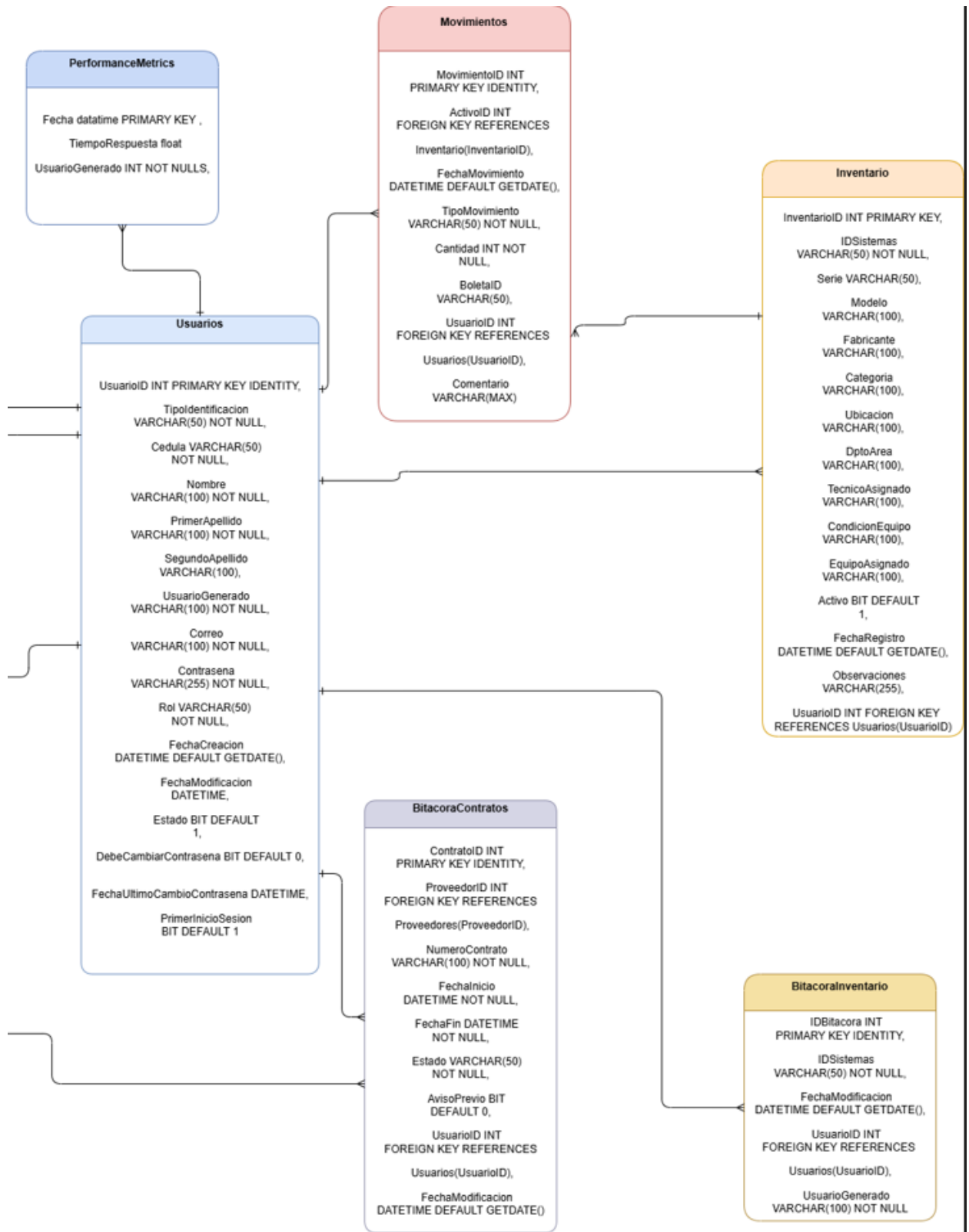


Figura 100: Modelo relacional



Diccionario de datos

Tabla 44: Tabla Usuarios

Columna	Tipo	Nulo	Predeterminado
UsuarioID (PK)	int (11)	No	
TipIdentificacion	varchar (50)	No	
Cedula	varchar (50)	No	
Nombre	varchar (191)	No	
PrimerApellido	varchar (191)	No	
SegundoApellido	varchar (191)	No	
UsuarioGenerado	varchar (100)	No	
Correo	varchar (191)	No	
Contraseña	varchar (255)	No	
Rol	varchar (50)	No	
FechaCreacion	timestamp	No	current_timestamp ()
FechaModificacion	timestamp	No	current_timestamp ()
Estado	varchar (50)	No	'Activo'

Tabla 45: Tabla Inventario

Columna	Tipo	Nulo	Predeterminado
InventarioID (PK)	int (11)	No	
IDSistemas	varchar (50)	No	
Serie	varchar (191)	No	
Modelo	varchar (191)	No	
Fabricante	varchar (191)	No	
Categoría	varchar (100)	No	
Ubicacion	varchar (255)	No	
DptoArea	varchar (100)	No	
TecnicoAsignado	int (11) (FK)	No	
CondicionEquipo	varchar (50)	No	
EquipoAsignado	varchar (100)	No	
Activo	varchar (10)	No	'Sí'
FechaRegistro	timestamp	No	current_timestamp ()
Observaciones	text	Sí	
UsuarioID (FK)	int (11)	No	

Tabla 46: Tabla Movimientos

Columna	Tipo	Nulo	Predeterminado
MovimientoID (PK)	int (11)	No	
ActivoID (FK)	int (11)	No	
FechaMovimiento	timestamp	No	current_timestamp ()
TipoMovimiento	varchar (50)	No	
Cantidad	int (11)	No	
BoletaID	int (11)	No	
UsuarioID (FK)	int (11)	No	
Comentario	text	Sí	
Producto	varchar (191)	No	
Serie	varchar (191)	No	
Modelo	varchar (191)	No	

Tabla 47: Tabla Contratos

Columna	Tipo	Nulo	Predeterminado
ContratoID (PK)	int (11)	No	
ProveedorID (FK)	int (11)	No	
NumeroContrato	varchar (50)	No	
FechaInicio	date	No	
FechaFin	date	No	
Estado	varchar (50)	No	
AvisoPrevio	int (11)	No	
UsuarioID (FK)	int (11)	No	
FechaModificacion	timestamp	No	current_timestamp ()

Tabla 48: Tabla Expedientes

Columna	Tipo	Nulo	Predeterminado
ExpedienteID (PK)	int (11)	No	
NombreUsuario	varchar (191)	No	
Descripcion	text	Sí	
FechaRegistro	timestamp	No	current_timestamp ()
UsuarioModificacion (FK)	int (11)	No	
FechaModificacion	timestamp	No	current_timestamp ()
TipoExpediente	varchar (100)	No	
FechaExpediente	date	No	

Tabla 49: Tabla Proveedores

Columna	Tipo	Nulo	Predeterminado
ProveedorID (PK)	int (11)	No	
NombreProveedor	varchar (191)	No	
Telefono	varchar (191)	Sí	
Email	varchar (191)	No	
ContratoID (FK)	int (11)	No	
TiempoContrato	int (11)	No	
CentroNegocio	varchar (100)	No	
FechaFinContrato	date	No	
PrecioContrato	decimal (10,2)	No	

Tabla 50: Tabla Bitácora Inventario

Columna	Tipo	Nulo	Predeterminado
IDBitacora (PK)	int (11)	No	
IDSistemas (FK)	varchar (50)	No	
FechaModificacion	timestamp	No	current_timestamp ()
Usuario	int (11)	No	
UsuarioGenerado	varchar (100)	No	

Tabla 51: Tabla Bitácora Contratos

Columna	Tipo	Nulo	Predeterminado
BitacoraContratoID (PK)	int (11)	No	
ContratoID (FK)	int (11)	No	
UsuarioID (FK)	int (11)	No	
Accion	varchar (50)	No	
Fecha	timestamp	No	current_timestamp ()
Descripcion	text	Sí	

Tabla 52: Tabla Bitácora

Columna	Tipo	Nulo	Predeterminado
BitacoraID (PK)	int (11)	No	

UsuarioID (FK)	int (11)	No	
Accion	varchar (50)	No	
Fecha	timestamp	No	current_timestamp ()
Descripcion	text	Sí	
Comentario	text	Sí	

Tabla 53: Tabla Bitácora Conexiones

Columna	Tipo	Nulo	Predeterminado
IDConexion (PK)	int (11)	No	
UsuarioID (FK)	int (11)	No	
FechaInicio	timestamp	No	current_timestamp ()
FechaFin	timestamp	Sí	
Estado	varchar (50)	No	

Desarrollo del sistema

Pantalla del Login

Figura 101: Login

FormLogin

Instacredit



Usuario	<input type="text"/>
Contraseña	<input type="password"/> 

Iniciar Sesión

Olvido Su Contraseña

Salir

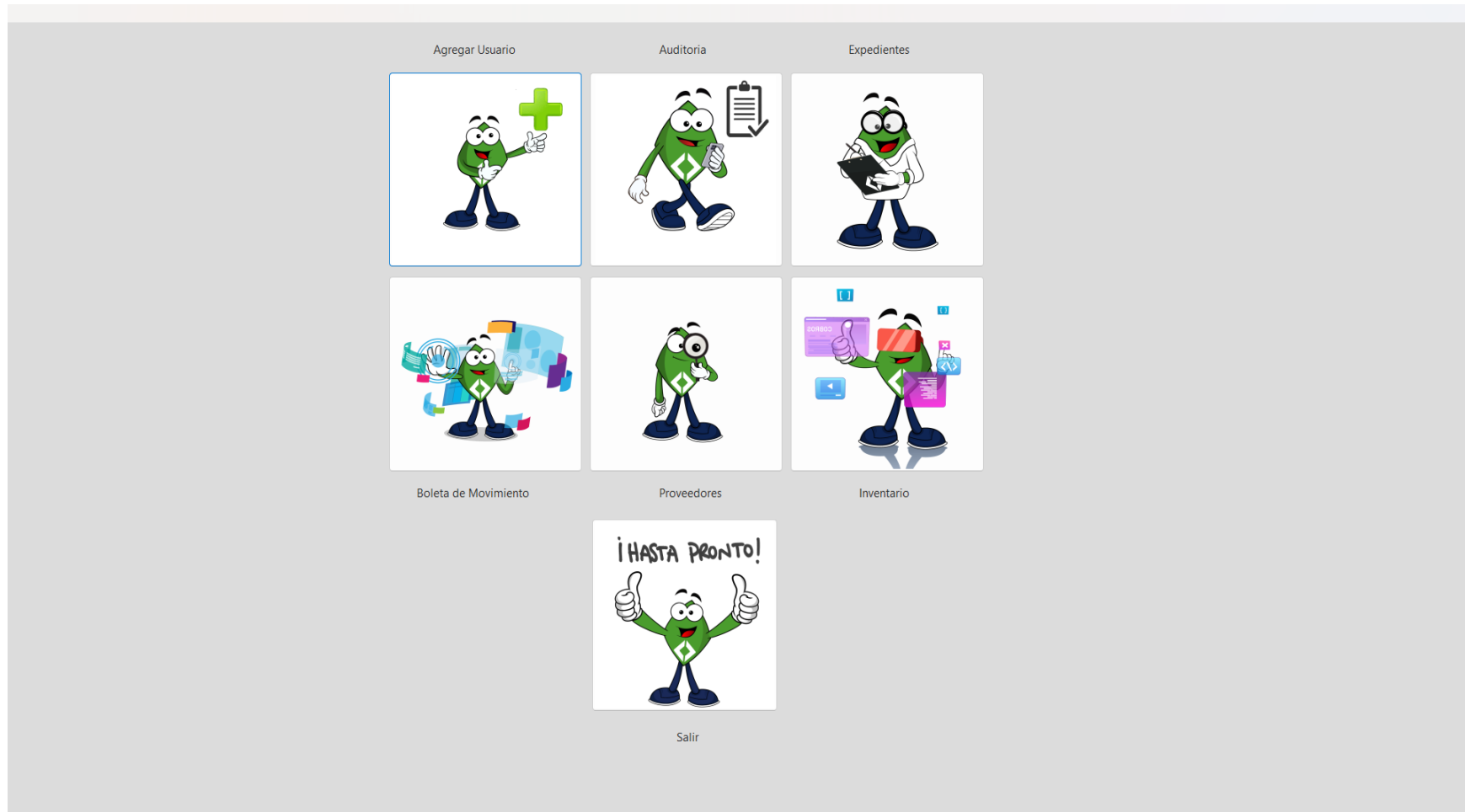


Soporte Técnico

Pantalla de Panel de control

Vista de Administrador.

Figura 102: Panel de control



Pantalla para agregar usuarios

Figura 103: Agregar Usuarios

Usuario

Tipo de Identificación Usuario


Cedula Email Estado Activo

Nombre Rol


Apellido 1

Apellido 2 Contraseña


UsuarioID	Tipoidentificacion	Cedula	Nombre	PrimerApellido	SegundoApellido	UsuarioGenerado	Correo	Contraseña	Rol
1	CEDULA DE IDENTIDAD	1-1561-0078	BRYAN JOSUE	ARGUELLO	SABORIO	bjarguellos	bjarguellos@instacredit.com	ccf208c91ce40991ceaca72d236ae81e876e13ddca75385aea29374d266ac98d	ADMINISTRAD
2	CEDULA DE IDENTIDAD	1-1646-0259	ULISES	VILLALOBOS	AGUILAR	uvillalobosa	uvillalobosa@instacredit.com	ccf208c91ce40991ceaca72d236ae81e876e13ddca75385aea29374d266ac98d	ESTANDAR
1002	CEDULA DE IDENTIDAD	9-0088-0938	MAURICIO MARTIN	TORRES	CERDA	mmtorresc	mmtorresc@instacredit.com	e9463a460d3da2fe90b595f64c879fdd58f7f76dabfa464332e3f0c109a5f648	ESTANDAR




Agregar Usuario




Buscar Usuario




Actualización de Datos



Inhabilitar Usuario



Limpiar



Salir

Figura 104: Ejemplo de cómo se agrega un usuario

Usuario

Tipo de Identificación: **CEDULA DE IDENTIDAD** ▾

Cedula: **1-1561-0079**

Nombre: **ROYNER FRANCISCO**

Apellido 1: **CASTRO**

Apellido 2: **DUARTE**

Usuario: **rfcastrod**

Email: **rfcastrod@instacredit.com**


Rol: ▾

Contraseña: Mostrar


Estado Activo

[Consultar Cedula](#)


	UsuarioID	Tipoidentificacion	Cedula	Nombre	PrimerApellido	SegundoApellido	UsuarioGenerado	Correo	Contraseña	Rol
▶	1	CEDULA DE IDENTIDAD	1-1561-0078	BRYAN JOSUE	ARGUELLO	SABORIO	bjarguellos	bjarguellos@instacredit.com	ccf208c91ce40991ceaca72d236ae81e876e13ddca75385aea29374d266ac98d	ADMINISTRAD
	2	CEDULA DE IDENTIDAD	1-1646-0259	ULISES	VILLALOBOS	AGUILAR	uvillalobosa	uvillalobosa@instacredit.com	ccf208c91ce40991ceaca72d236ae81e876e13ddca75385aea29374d266ac98d	ESTANDAR
	1002	CEDULA DE IDENTIDAD	9-0088-0938	MAURICIO MARTIN	TORRES	CERDA	mmtorresc	mmtorresc@instacredit.com	e9463a460d3da2fe90b595f64c879fdd587f7f6dabfa464332e3f0c109a5f648	ESTANDAR
*										




Agregar Usuario




Buscar Usuario




Actualización de Datos



Inhabilitar Usuario



Limpiar



Salir

Figura 105: Ejemplo de cómo buscar usuarios.

Usuario

Tipo de Identificación: **CEDULA DE IDENTIDAD** (dropdown)

Cedula: **1-1561-0079**

Nombre: **ROYNER FRANCISCO**

Apellido 1: **CASTRO**

Apellido 2: **DUARTE**

Usuario: **rfgastrod**

Email: **rfgastrod@instacredit.com**

Rol: (dropdown)

Contraseña: (input) Mostrar

Estado Activo


Consultar Cedula

UsuarioID	Tipoidentificacion	Cedula	Nombre	PrimerApellido	SegundoApellido	UsuarioGenerado	Correo	Contraseña	Rol
1	CEDULA DE IDENTIDAD	1-1561-0078	BRYAN JOSUE	ARGUELLO	SABORIO		rfgastrod@instacredit.com	ccf208c91ce40991ceaca72d236ae81e876e13ddca75385aea29374d266ac98d	ADMINISTRADOR
2	CEDULA DE IDENTIDAD	1-1646-0259	ULISES	VILLALOBOS	AGUILAR		rfgastrod@instacredit.com	ccf208c91ce40991ceaca72d236ae81e876e13ddca75385aea29374d266ac98d	ESTANDAR
1002	CEDULA DE IDENTIDAD	9-0088-0938	MAURICIO MARTIN	TORRES	CERDA		rfgastrod@instacredit.com	e9463a460d3da2fe90b595f64c879fd58f7f76dabfa464332e3f0c109a5f648	ESTANDAR


Buscar Usuario

Ingrese el criterio de búsqueda (Cédula, Nombre, Apellido, Correo, etc):


Aceptar Cancelar




Agregar Usuario




Buscar Usuario




Actualización de Datos



Inhabilitar Usuario



Limpiar



Salir

Pantalla reportes Auditoría

En esta pantalla, se puede filtrar por fecha o reporte específico.

Figura 106: Pantalla de Reportes para Auditoría.

FormReporteAuditoria


Fecha de Inicio


Fecha Final


Correo

Bitacoras

- Bitácora de Usuarios
- Bitácora de Conexiones
- Bitácora de Contratos
- Bitácora de Inventarios

 Descargar Reportes

 Enviar Reportes


 Salir

Pantalla de expedientes

En esta pantalla, se muestran los expedientes creados para el personal.

Figura 107: Pantalla Expedientes

FormExpedientes


viernes , 28 de febrero de 2025

Busqueda


Expediente ID

Tipo de Expediente


Seleccione el Usuario

Descripción


	ExpedienteID	Nombre Usuario	Descripción	Fecha de Registro	Usuario Modificación	Fecha de Modificación	Tipo de Expediente
▶	6	BRYAN JOSUE ARGUELLO SABORIO	Prueba 1	25/1/2025 17:48	1	18/2/2025 13:20	Seguimiento
	7	BRYAN	Prueba 2	25/1/2025 17:48	1		Amonestacion Verba
	8	BRYAN	Prueba	25/1/2025 17:57	1		Seguimiento
	9	BRYAN	Prueba 3	26/1/2025 09:40	1		Seguimiento
	1004	BRYAN ARGUELLO SABORIO	Pruebas	26/1/2025 14:48	1		Seguimiento
	1005	BRYAN ARGUELLO SABORIO	w	26/1/2025 14:49	1		Seguimiento
	1006	BRYAN JOSUE ARGUELLO SABORIO	Prueba 2	27/1/2025 18:57	1	19/2/2025 11:34	Despido Sin Respon




Actualizar



Limpiar



Agregar

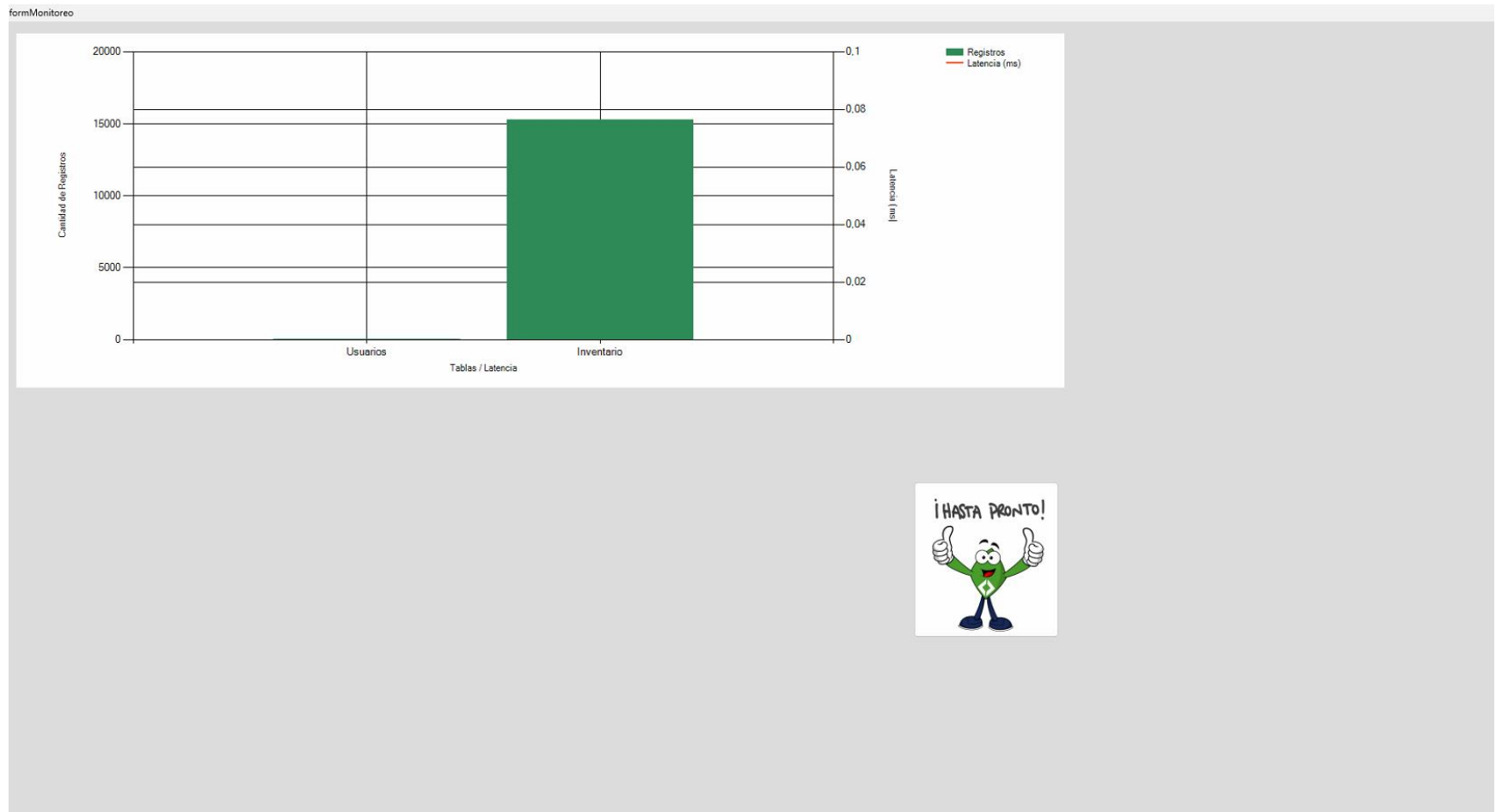


Salir

Pantalla de monitoreo

En esta pantalla se puede monitorear el estado de salud de las tablas y la latencia que se tiene entre la aplicación y la base de datos SQL.

Figura 108: Pantalla de monitoreo



Pantalla de proveedores

En esta pantalla, se muestran los proveedores con los que cuenta la empresa.

Figura 110: Pantalla proveedores

FormGestionarProveedores

ProveedorID	NombreProveedor	Telefono	Email	ContratoID	TiempoContrato	CentroNegocio	FechaFinContrato	PrecioContrato	Estado	Fec
1	Telecable - 100 Mbps	2222-2222	telecable100mbps@proveedor.com	TC1001	3	CANADA	31/12/2023	500,00	<input checked="" type="checkbox"/>	
2	Telecable - 200 Mbps	2222-2223	telecable200mbps@proveedor.com	TC2001	5	HEREDIA	29/2/2024	800,00	<input checked="" type="checkbox"/>	
3	Tigo - 50 Mbps	2222-3333	tigo50mbps@proveedor.com	TG5001	2	AV2	30/11/2023	300,00	<input checked="" type="checkbox"/>	
4	Tigo - 100 Mbps	2222-3334	tigo100mbps@proveedor.com	TG1001	1	LIMON	31/10/2023	600,00	<input checked="" type="checkbox"/>	
5	Claro - 60 Mbps	2222-4444	claro60mbps@proveedor.com	CL6001	12	SQUIRRES	30/9/2024	400,00	<input checked="" type="checkbox"/>	
6	Claro - 120 Mbps	2222-4445	claro120mbps@proveedor.com	CL1201	12	SIGMA	30/9/2024	700,00	<input checked="" type="checkbox"/>	
7	Kolbi - 30 Mbps	2222-5555	kolbi30mbps@proveedor.com	KB3001	11	GUAPILES	31/8/2024	200,00	<input checked="" type="checkbox"/>	
8	Kolbi - 60 Mbps	2222-5556	kolbi60mbps@proveedor.com	KB6001	8	Centro Negocio Heredia	31/5/2024	450,00	<input checked="" type="checkbox"/>	
9	Telecable - 500 Mbps	2222-2224	telecable500mbps@proveedor.com	TC5001	8	MATUTE	31/5/2024	1200,00	<input checked="" type="checkbox"/>	
10	Tigo - 200 Mbps	2222-3335	tigo200mbps@proveedor.com	TG2001	9	PUNTARENAS	30/6/2024	1000,00	<input checked="" type="checkbox"/>	
11	Claro - 200 Mbps	2222-4446	claro200mbps@proveedor.com	CL2001	10	PLAZA AMERICA	31/7/2024	900,00	<input checked="" type="checkbox"/>	
12	Kolbi - 100 Mbps	2222-5557	kolbi100mbps@proveedor.com	KB1001	1	DESAMPARADOS	31/10/2023	350,00	<input checked="" type="checkbox"/>	
13	Telecable - 100 Mbps	2222-2222	telecable100mbps@proveedor.com	TC1001	12	OROTINA	30/9/2024		<input checked="" type="checkbox"/>	
14	TELECABLE - 200 MBPS	2222-2223	TELECABLE200MBPS@PROVEEDOR.COM	TC2001	24	PUNTARENAS	22/2/2025	560,00	<input checked="" type="checkbox"/>	

Proveedor ID

Nombre Proveedor

Telefono

Email

Contrato ID


Centro De Negocio

Fecha Fin Contrato


Precio De Contrato

Tiempo De Contrato


Buscar




Formulario




Inhabilitar




Actualizar



Agregar



Limpiar



Salir

Figura 111: Pantalla para agregar nuevos proveedores

Proveedor ID	<input type="text"/>	Centro De Negocio	<input type="text"/>
Nombre Proveedor	<input type="text"/>	Fecha Fin Contrato	28/ 2/2025
Telefono	<input type="text"/>	Precio De Contrato	<input type="text"/>
Email	<input type="text"/>	Tiempo De Contrato	<input type="text"/>
Contrato ID	<input type="text"/>		

Agregar Proveedor	Limpiar	Salir

Pantalla de inventarios

En esta pantalla se observa el inventario.

Figura 112: Pantalla de inventarios

Inventario

ID Sistemas

Serie

Modelo

Fabricante

Categoria

Condicion Equipo

DptoArea

Tecnico Asignado

Numero de Activo

Numero de Activo


Equipo Asignado


Fecha Registro **viernes , 28 de febrero de 2025**


Observaciones


InventarioID	IDSistemas	Serie	Modelo	Fabricante	Categoria	Ubicacion	DptoArea
1	ABC-001	INC21036302157	MODEL-A	FAB-XYZ	CATEGORIA-1	MATUTE	SISTEMAS
2	ABC-002	INC21036302146	MODEL-B	FAB-ABC	CATEGORIA-2	UBICACION-2	DPTO-2
3	ABC-003	INC210363022149	MODEL-C	FAB-123	CATEGORIA-3	MATUTE	SISTEMAS
4	M601	INC2015301541	KMO-506	KLIPXTREME	MOUSE	EDIFICIO GAMA	CONSORCIO JURIDICO
5	DELL171	DGF3831	INSPIRON 5100		PORTÁTIL	BAJA	
6	DELL172	JZD0Y51	INSPIRON 600M		PORTÁTIL	BAJA	
7	DELL201	20F0Y51	INSPIRON 600M		PORTÁTIL	BAJA	
8	DELL220	6HMZH91	INSPIRON 630M		PORTÁTIL	BAJA	
9	DELL221	GJMZH91	INSPIRON 630M		PORTÁTIL	BAJA	
10	DELL222	JJMZH91	INSPIRON 630M		PORTÁTIL	BAJA	
11	DELL223	5K0LX41	INSPIRON 1150		PORTÁTIL	BAJA	
12	DELL224	9P7ST21	INSPIRON 5100		PORTÁTIL	BAJA	
13	DELL225	8L0LX41	INSPIRON 1150		PORTÁTIL	BAJA	
14	DELL241	16F62B1	INSPIRON 1150		PORTÁTIL	BAJA	


Buscar





Insertar


Actualizar


Inhabilitar


SubirArchivo


Limpiar



Salir

Figura 113: Opción de subir archivos CSV para actualizar inventarios

The screenshot shows a software interface with a file selection dialog box open over a table. The dialog box is titled 'Seleccionar archivo CSV' and shows a file named 'CSV para subir al Inventario' selected. The table below has columns for 'Id', 'Modelo', 'Serial', 'Marca', 'Categoría', 'Ubicación', and 'DptoArea'. At the bottom, there is a toolbar with icons for 'Insertar', 'Actualizar', 'Inhabilitar', 'SubirArchivo', 'Limpiar', and 'Salir'. The 'SubirArchivo' icon is highlighted.

Id	Modelo	Serial	Marca	Categoría	Ubicación	DptoArea
8	DELL220	6HMZH91	INSPIRON	INSPIRON 630M	BAJA	
9	DELL221	GJMZH91	INSPIRON	INSPIRON 630M	BAJA	
10	DELL222	JJMZH91	INSPIRON	INSPIRON 630M	BAJA	
11	DELL223	5K0LX41	INSPIRON	INSPIRON 1150	BAJA	
12	DELL224	9P7ST21	INSPIRON	INSPIRON 5100	BAJA	
13	DELL225	8L0LX41	INSPIRON	INSPIRON 1150	BAJA	
14	DELL241	16F62B1	INSPIRON	INSPIRON 1150	BAJA	

Referencias

- "Análisis Diseño y Construcción de una Aplicación Web para la Gestión de Activos de TI" | Portelles Cobas, R. (2015). *Xabal Repxos*. Obtenido de Xabal Repxos: <https://repositorio.uci.cu/jspui/handle/ident/9010>
- "Diseño de Propuesta de Modelo de Gestión de Activos basado en la Norma ISO 55000 y un Sistema Integrado de Gestión del Espacio de Trabajo (IWMS)" | Campos Zúñiga. (2014). *Repositorio Tec*. Obtenido de Repositorio Tec: <https://repositorio.uci.cu/jspui/handle/ident/9010>
- ¿Estudio de factibilidad: ¿por qué es importante para un proyecto? | Ingeniería VQ. (2021). *Vqingenieria.com*. Obtenido de Vqingenieria.com: <https://www.vqingenieria.com/los-siete-pasos-para-realizar-un-estudio-de-factibilidad-con-exito>
- ¿Qué es la arquitectura de datos? | IBM. (2024). *Ibm.com*. Obtenido de Ibm.com: <https://www.ibm.com/es-es/topics/data-architecture>
- ¿Qué es la arquitectura orientada a servicios? | Amazon. (s.f.). *Amazon.com*. Obtenido de Amazon.com: <https://aws.amazon.com/es/what-is/service-oriented-architecture/>
- ¿Qué es la gestión de requisitos? | Ibm.com. (2024). *Ibm.com*. Obtenido de Ibm.com: <https://www.ibm.com/mx-es/topics/what-is-requirements-management>
- ¿Qué es la gestión del ciclo de vida de los activos (ALM)? Etapas y estrategia | eMaint. (2024). *eMaint*. Obtenido de eMaint: <https://www.emaint.com/es/blog-what-is-asset-lifecycle-management/>
- ¿Qué es la gestión del ciclo de vida de los activos? | IBM. (2024). *Ibm.com*. Obtenido de Ibm.com: <https://www.ibm.com/es-es/topics/asset-lifecycle-management>
- ¿Qué es la integración de bases de datos? Descripción general y beneficios | Naeem, T. (2019). *Astera*. Obtenido de Astera: <https://www.astera.com/es/type/blog/database-integration/>
- ¿Qué es la metodología Kanban y cómo funciona? | Martins, J. (2025). *Asana*. Obtenido de Asana: <https://asana.com/es/resources/what-is-kanban>
- ¿Qué es la metodología? Definición características y tipos? | Pinterest. (s.f.). *Pinterest*. Obtenido de Pinterest: <https://es.pinterest.com/pin/749567931723639237/>
- ¿Qué es la Programación orientada a objetos? | Edteam. (2020). *EDteam*. Obtenido de EDteam: <https://ed.team/comunidad/que-es-la-programacion-orientada-a-objetos>
- ¿Qué es la SOA (arquitectura orientada a servicios)? | Ibm.com. (2023). *Ibm.com*. Obtenido de Ibm.com: <https://www.ibm.com/mx-es/topics/soa>
- ¿Qué es Microsoft SQL Server y para qué sirve? | Pérez, S. D. (2021). *Intelequia*. Obtenido de Intelequia: <https://intelequia.com/es/blog/post/qu%C3%A9-es-microsoft-sql-server-y-para-qu%C3%A9-sirve>
- ¿Qué es un sistema de información de marketing en la actualidad? | Migallón, L. (2021). *Witei*. Obtenido de Witei: <https://get.witei.com/es/articulos/que-es-un-sistema-de-informacion-de-marketing-en-la-actualidad/>

- ¿Qué es un sistema de procesamiento de transacciones(TPS)?|IBM. (2024). *Ibm.com*. Obtenido de Ibm.com: <https://www.ibm.com/mx-es/topics/transaction-processing-system>
- ¿Qué es una base de datos de código abierto?|Purestorage. (2021). *Purestorage.com*. Obtenido de Purestorage.com: <https://www.purestorage.com/la/knowledge/what-is-an-open-source-database.html>
- ¿Qué es una base de datos en la nube?|Ibm.com. (2023). *Ibm.com*. Obtenido de Ibm.com: <https://www.ibm.com/es-es/topics/cloud-database>
- ¿Qué son los Digital assets management (DAM) o sistema de gestión de activos digitales?|ESIC. (2019). *Esic.edu*. Obtenido de Esic.edu: <https://www.esic.edu/rethink/comercial-y-ventas/digital-assets-management>
- ¿Qué son los lenguajes tipados y no tipados? (Explicación sencilla)|Ed.team. (s.f.). *EDteam*. Obtenido de <https://ed.team/blog/que-son-los-lenguajes-tipados-y-no-tipados-explicacion-sencilla>
- ¿Qué son los paradigmas de programación?|.Ed.team. (s.f.). *¿Qué son los paradigmas de programación?* Obtenido de ¿Qué son los paradigmas de programación?: <https://ed.team/blog/que-son-los-paradigmas-de-programacion-f57de068-6426-4332-b5dd-dbdcff1aaec>
- 3digits. (2021). *3digits.es*. Obtenido de <https://learn.microsoft.com/es-es/visualstudio/ide/visual-studio-multi-targeting-overview?view=vs-2022>
- 4 steps to create a basic windows forms application|Apriorit. (2016). *Apriorit*. Obtenido de Apriorit: <https://www.apriorit.com/dev-blog/426-basic-windows-forms-explained>
- Algoritmos en la programación |Ed.team. (2021). *Ed.team*. Obtenido de .Ed.team: <https://ed.team/cursos/algoritmos>
- Análisis del uso de herramientas informáticas para la gestión de activos empresariales en las empresas del sector alimenticio registradas en la Cámara de la Pequeña y Mediana Empresa de Pichincha|Chavez Liquinchana. (2020). *Universidad Andina Simón Bolívar*. Obtenido de Universidad Andina Simón Bolívar: <https://repositorio.uasb.edu.ec/handle/10644/7721>
- Apache CouchDB|Apache.org. (s.f.). *Apache.org*. Obtenido de Apache.org: <https://couchdb.apache.org/>
- Arquitectura Cliente-Servidor|Programación reactiva. (s.f.). *Programación reactiva*. Obtenido de Programación reactiva: <https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/cliente-servidor>
- Arquitectura de componentes|Medium. (2024). *Medium*. Obtenido de Medium: <https://vanessamarely.medium.com/arquitectura-de-componentes-e48816925d51>
- Arquitectura de datos|Datdata.com. (2024). *Datdata.com*. Obtenido de Datdata.com: <https://www.datdata.com/blog/arquitectura-de-datos>
- Arquitectura de microservicios: una solución para la escalabilidad y flexibilidad de aplicaciones|MyTaskPanel Consulting. (2023). *MyTaskPanel Consulting*. Obtenido de MyTaskPanel Consulting: <https://www.mytaskpanel.com/arquitectura-de-microservicios-escalabilidad-y-flexibilidad/>

- Base de Datos Descentralizada | Coinex.com. (s.f.). *Coinex.com*. Obtenido de Coinex.com:
<https://www.coinex.com/es/academy/detail/1239-decentralized-database?pld=1>
- Base de datos federada | Db2 11.1. (2023). *Ibm.com*. Obtenido de Ibm.com:
<https://www.ibm.com/docs/es/db2/11.1?topic=systems-federated-database>
- Base de datos Open Source | ITD Consulting. (2022). *ITD Consulting*. Obtenido de ITD Consulting:
<https://itdconsulting.com/base-de-datos/mysql-maria-db/base-de-datos-open-source/>
- Campos Zúñiga. (2014). *Repositorio Tec*. Obtenido de Repositorio Tec:
<https://repositoriotec.tec.ac.cr/handle/2238/3345>
- Características del lenguaje de programación | ISC. (s.f.). *ISC*. Obtenido de ISC: <https://tutoriales-isc.blogspot.com/2018/05/311-caracteristicas-del-lenguaje-de.html>
- Ciclo de Vida de la Información | Edu.ar. (s.f.). *Edu.ar*. Obtenido de Edu.ar:
https://educacion.sanjuan.edu.ar/mesj/LinkClick.aspx?fileticket=_jqHbG_qr9w%3D&tabid=679&mid=1746
- Ciclo de vida de un sistema de información | Guerrero, A. C. (2015). *gestiopolis*. Obtenido de gestiopolis:
<https://www.gestiopolis.com/ciclo-de-vida-de-un-sistema-de-informacion/>
- Cómo redactar planes de gestión de riesgos y por qué hacerlo es importante | Escuela Europea de Excelencia. (2022). *Escuela Europea de Excelencia*. Obtenido de Escuela Europea de Excelencia:
<https://www.escolaeuropeaexcelencia.com/2022/11/como-redactar-planes-de-gestion-de-riesgos-y-por-que-hacerlo-es-importante/>
- Cómo redactar Requerimientos de Usuario | Sandoval, J. (2022). *QbD.Digital*. Obtenido de QbD.Digital:
<https://qbd.digital/como-redactar-requerimientos-de-usuario/>
- Comparativa de 5 bases de datos open source | IONOS Digital Guide. (s.f.). *IONOS Digital Guide*. Obtenido de IONOS Digital Guide: <https://www.ionos.com/es-us/digitalguide/hosting/cuestiones-tecnicas/comparativa-de-bases-de-datos-open-source/>
- Conteo Cíclico | Bot Verification. (s.f.). *Com.mx*. Obtenido de Com.mx: <https://bind.com.mx/blog/control-de-inventarios/tecnicas-y-metodos-para-el-control-de-inventarios>
- Crud con Programación Funcional en Windows Forms C# | Zambrana, J. (2021). *Crud con Programación Funcional en Windows Forms C*. Obtenido de Crud con Programación Funcional en Windows Forms C: <https://www.youtube.com/watch?v=ljrwhyQxO8k>
- de Caldas, J., Ardila García, & Sánchez Talero. (2015). *xabal repxos*. Obtenido de xabal repxos:
<https://repository.udistrital.edu.co/items/1a217d91-857a-47c2-8d1b-7416a97a3419>
- Definición de casos de uso | IBM Product Master 12.0.0. (2025). *IBM Product Master 12.0.0*. Obtenido de IBM Product Master 12.0.0: <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>
- Definición de enfoques de investigación | El portal de la tesis. (Ucol.mx). *El portal de la tesis*. Obtenido de Ucol.mx: <https://recursos.ucol.mx/tesis/investigacion.php>

- DEFINICIÓN DE SISTEMA DE INFORMACIÓN | sicunisucre. (2014). *sicunisucre2014*. Obtenido de sicunisucre2014: <https://sicunisucre2014.wordpress.com/2014/10/16/definicion-de-sistema-de-informacion/>
- Diagrama de casos de Uso | Fandom Inc. (s.f.). *ADSI Wiki*. Obtenido de ADSI Wiki: https://adsi.fandom.com/wiki/Diagrama_de_casos_de_Uso
- Diagrama de componentes UML | miro. (s.f.). <https://miro.com/>. Obtenido de <https://miro.com/>: <https://miro.com/es/diagrama/que-es-diagrama-componentes-uml/>
- Diagrama del problema solución y aspectos tecnológicos | GESTIÓN DE RIESGOS. (s.f.). *Scielo.cl*. Obtenido de Scielo.cl: https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000300135
- Diccionario de Marketing e Internet en general, Q. (s.f.). Obtenido de <https://www.appyweb.es/diccionario/queue/>
- Diseño de investigación. Elementos y características | Muguira, A. (2018). *QuestionPro*. Obtenido de QuestionPro: <https://www.questionpro.com/blog/es/disenio-de-investigacion/>
- Dos siglos de historia del lenguaje de programación | RedaccionONOFF, P. (2022). *Dos siglos de historia del lenguaje de programación*. Obtenido de Dos siglos de historia del lenguaje de programación: <https://www.revistaonoff.es/lenguaje-de-programacion-una-historia-con-mas-de-200-anos/>
- Ejemplo de pantallas de VB.NET Windows Forms | VB Net Forms. (s.f.). *Tutorialspoint.com*. Obtenido de Tutorialspoint.com: https://www.tutorialspoint.com/vb.net/vb.net_forms.htm
- El futuro de la arquitectura de software: tendencias a tener en cuenta | Aicad Business School. (2024). *Aicad Business School*. Obtenido de Aicad Business School: <https://www.aicad.es/arquitectura-de-software>
- Enfoque tipo diseño y método de investigación [Aclarando conceptos] | Samaniego, G. (2022). *Asesoría de Tesis*. Obtenido de Asesoría de Tesis: <https://miasesordetesis.com/enfoque-tipo-diseno-metodo-de-investigacion/>
- Estudio de Factibilidad | de Luis R Castellanos. (2009). *Desarrollo de Sistemas de Información*. Obtenido de Desarrollo de Sistemas de Información: <https://desarrollodesistemas.wordpress.com/2009/07/05/estudio-de-factibilidad/>
- Explorando los lenguajes de interrogación de bases de datos: El estándar ANSI SQL | Cheque Formación. (2024). *Cheque Formación*. Obtenido de Cheque Formación: <https://chequeformacion.es/informatica-y-tecnologia/lenguajes-de-interrogacion-de-bases-de-datos-estandar-ansi-sql/>
- Factibilidad Operativa | Economipedia.com. (s.f.). *Economipedia.com*. Obtenido de Economipedia.com: <https://economipedia.com/definiciones/factibilidad-operativa.html>
- FIFO y LIFO: técnicas de gestión de la carga, N. S. (2017). *Noega Systems*. Obtenido de Noega Systems: <https://www.noegasystems.com/blog/logistica/fifo-y-lifo-tecnicas-de-almacenaje>

- Fuentes de información - Qué son, t. y. (s.f.). *concepto.de*. Obtenido de concepto.de:
<https://concepto.de/fuentes-de-informacion/>
- Gestión digital de expedientes en Sysco | Salas-Vargas, L. E. (2023). *RepositorioTEC*. Obtenido de RepositorioTEC: <https://repositoriotec.tec.ac.cr/handle/2238/14489>
- Glosario | {Docu Ware Europe GmbH}. (2025). *Docuware.com*. Obtenido de Docuware.com:
<https://start.docuware.com/es/glosario-de-terminos/expedientes-de-empleados>
- Grupo Comunicar – Web Oficial del Grupo Comunicar | Grupocomunicar. (s.f.). *Grupocomunicar.com*. Obtenido de Grupocomunicar.com: <http://grupocomunicar.com>
- Hagamos una copia de seguridad de tu PC | {Microsoft}. (2023). *{Microsoft}*; Obtenido de {Microsoft};:
<https://www.microsoft.com/es-es/windows/end-of-support?r=1>
- Historia de las Bases de Datos: Un Viaje a través del Tiempo | Mosquera, J. (2024). *Jhon Mosquera*. Obtenido de Jhon Mosquera: <https://jhonmosquera.com/bases-de-datos-historia/>
- IBM. (2025). *¿Qué es la gestión de activos digitales?* Obtenido de ¿Qué es la gestión de activos digitales?:
<https://www.ibm.com/mx-es/topics/digital-asset-management>
- IBM. (2025). *¿Qué es la gestión de proveedores?* Obtenido de ¿Qué es la gestión de proveedores?:
<https://www.ibm.com/es-es/topics/supplier-management>
- IBM. (2025). *Gestión de registros de usuarios*. Obtenido de Gestión de registros de usuarios:
<https://www.ibm.com/docs/es/sva/10.0.8?topic=configuration-managing-user-registries>
- IBM. (2025). *Optimice las auditorías internas*. Obtenido de Optimice las auditorías internas:
<https://www.ibm.com/es-es/products/openpages/internal-audit>
- IBM Security QRadar Log Insights | Ibm.com. (2024). *Ibm.com*. Obtenido de Ibm.com:
<https://www.ibm.com/docs/es/security-qradar/log-insights/saas?topic=sources-microsoft-sql-server-source>
- Implementación de EOQ en la gestión de inventario. (s.f.). *Fastercapital.com*. Obtenido de Fastercapital.com: <https://fastercapital.com/es/tema/m%C3%A9todos-para-determinar-el-valor-de-liquidaci%C3%B3n.html>
- Ingeniería de Requisitos (Software Requirements) | We're always in the KnowGarden. (2015). *We're always in the KnowGarden*. Obtenido de We're always in the KnowGarden:
<https://knowgarden.wordpress.com/2015/01/27/ingenieria-de-requisitos-software-requirements/>
- Interfaz de programación | appleute. (2021). *appleute*. Obtenido de appleute:
<https://www.appleute.de/es/biblioteca-para-desarrolladores-de-aplicaciones/interfaz-de-programacion/>
- Introducción al desarrollo de aplicaciones de escritorio con.net | Eafranco.com. (s.f.). *Eafranco.com*. Obtenido de Eafranco.com: Introducción al desarrollo de aplicaciones de escritorio con .NET

Introducción: ¿Qué es un sistema ERP? | Perfil, V. (s.f.). *Blogspot.com*. Obtenido de Blogspot.com: <https://sistemainformacionerp.blogspot.com/2017/10/que-es-un-sistema-erp.html>

Inventario de activos y gestión de la seguridad en SCI | Incibe.es. (2016). *INCIBE*. Obtenido de INCIBE: <https://www.incibe.es/incibe-cert/blog/inventario-activos-y-gestion-seguridad-sci>

ISO 55000 | ISO. (2014). *ISO55000*. Obtenido de ISO55000: <https://www.iso.org/obp/ui#iso:std:iso:55000:ed-1:v2:es>

Lenguajes de Alto nivel vs de Bajo nivel | EDteam. (s.f.). *EDteam*. Obtenido de EDteam: <https://ed.team/comunidad/lenguajes-de-alto-nivel-vs-de-bajo-nivel-b8806b09-73ac-4d83-9047-b487dae2cab2>

Lenguajes de Programación | Monterde, L. d., & Marín. (s.f.). *Lenguajes de Programación*. Obtenido de Lenguajes de Programación: https://repositorio-uapa.cuaieed.unam.mx/repositorio/moodle/pluginfile.php/2655/mod_resource/content/1/UAPA-Lenguajes-Programacion/index.html

MAPA CONCEPTUAL - GESTIÓN DE COMUNICACIÓN EN LOS PROYECTOS | Castellanos, K. (s.f.). *MAPA CONCEPTUAL - GESTIÓN DE COMUNICACIÓN EN LOS PROYECTOS*. Obtenido de MAPA CONCEPTUAL - GESTIÓN DE COMUNICACIÓN EN LOS PROYECTOS: <https://view.genially.com/63703087436ebb00181b09ff/interactive-content-mapa-conceptual-gestion-de-comunicacion-en-los-proyectos>

Marco metodológico: Qué es y cómo realizarlo | Ortega, C. (2024). *QuestionPro*. Obtenido de QuestionPro: <https://www.questionpro.com/blog/es/marco-metodologico/>

Matriz de riesgos: cómo evaluar los riesgos para lograr el éxito del proyecto [2025] | Asana. (2025). *Asana*. Obtenido de Asana: <https://asana.com/es/resources/risk-matrix-template>

Método ABC | Control de inventarios. (s.f.). *Control de inventarios*. Obtenido de Control de inventarios: <https://controlinventarios.wordpress.com/2023/01/17/metodo-abc-2/>

Método ABC de inventarios en almacén: que es | AR-Racking. (s.f.). *Ar-racking.com*. Obtenido de Ar-racking.com: <https://www.ar-racking.com/es/blog/metodo-abc-de-inventarios-en-almacen-origen-caracteristicas-y-ventajas/>

Metodologías de desarrollo de software: ¿qué son? | Santanderopenacademy.com. (s.f.). *Santanderopenacademy.com*. Obtenido de Santanderopenacademy.com: <https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>

Metodologías de desarrollo de software | de Licenciatura en Sistemas y Computación, T. (s.f.). *Edu.ar*. Obtenido de Edu.ar: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>

Microsoft. (12 de 12 de 2023). *Microsoft*. Obtenido de <https://learn.microsoft.com/es-es/visualstudio/ide/visual-studio-multi-targeting-overview?view=vs-2022>

Microsoft. (2024). *Microsoft*. Obtenido de <https://visualstudio.microsoft.com/es/#vs-section>

- MODELO DE GESTIÓN DE ACTIVOS PARA EL ÁREA DE TECNOLOGÍAS DE LA | Aguilar Barboza. (2024). *Universidad de Costa Rica*. Obtenido de Universidad de Costa Rica.: <https://www.kerwa.ucr.ac.cr/server/api/core/bitstreams/f40418e3-46ae-4157-8114-abede406e1bd/content>
- Monolitica | Amazon.com, A. (s.f.). *Amazon.com*. Obtenido de Amazon.com: <https://aws.amazon.com/es/compare/the-difference-between-monolithic-and-microservices-architecture/#:~:text=Una%20arquitectura%20monol%C3%ADtica%20es%20un,de%20datos%20dentro%20del%20sistema.>
- Mora Matamoros. (2022). *Repositorio TEC*. Obtenido de Repositorio TEC: <https://repositoriotec.tec.ac.cr/handle/2238/14008>
- Programación Orientada a Eventos: Introducción y conceptos | Canelo, M. M. (2024). *Profile Software Services*. Obtenido de Profile Software Services: <https://profile.es/blog/programacion-orientada-a-eventos/>
- Qué es MySQL: Características y ventajas | Robledano, A. (2019). *OpenWebinars.net*. Obtenido de OpenWebinars.net: <https://openwebinars.net/blog/que-es-mysql/>
- Recalde Manrique, & Andre Walter. (2022). *PUCP*. Obtenido de PUCP: <https://tesis.pucp.edu.pe/items/c4a708e9-cb19-43bc-8b02-0a0f0c8af5d5>
- Requerimientos en el desarrollo de software y aplicaciones | Northware. (2022). *Northware*. Obtenido de Northware: <https://www.northware.mx/blog/requerimientos-en-el-desarrollo-de-software-y-aplicaciones/>
- Requerimientos funcionales | Pmoinformatica.com. (s.f.). *Pmoinformatica.com*. Obtenido de Pmoinformatica.com: <https://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>
- Requerimientos funcionales: Ejemplos | Pmoinformatica.com. (s.f.). *Pmoinformatica.com*. Obtenido de Pmoinformatica.com: <https://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>
- Requerimientos no funcionales: Ejemplos | Pmoinformatica.com. (s.f.). *Pmoinformatica.com*. Obtenido de Pmoinformatica.com: <https://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>
- Requerimientos no funcionales | Pmoinformatica.com. (s.f.). *Pmoinformatica.com*. Obtenido de Pmoinformatica.com: <https://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>
- Salas Vargas. (2023). *Repositorio TEC*. Obtenido de Repositorio TEC: <https://repositoriotec.tec.ac.cr/handle/2238/14489>
- Sistema de control de procesos de negocio (BPM): Ejemplos y características | Sierra, & Yhorman. (2022). *Legaltech*. Obtenido de Legaltech: <https://blog.lemontech.com/sistema-control-procesos-negocio/>

- Sistema EOQ o Modelo de Wilson en almacén | Ar-racking.com. (2021). *AR-Racking*. Obtenido de AR-Racking: <https://www.ar-racking.com/es/blog/sistema-eoq-o-modelo-de-wilson-en-almacen-gestion-de-stocks/>
- Sistema para el control de activos informáticos de la facultad 4" | Herrera Vive, Y. (2015). *Xabal Repxos*. Obtenido de Xabal Repxos: <https://repositorio.uci.cu/jspui/handle/ident/9010>
- Sistemas de Información Ejecutiva (EIS) | Sinnexus. (s.f.). *Sinnexus.com*. Obtenido de Sinnexus.com: https://www.sinnexus.com/business_intelligence/sistemas_informacion_ejecutiva.aspx
- Sistemas de Información Gerencial - Objetivos, B. y.-i. (s.f.). *Tecnologias-informacion.com*. Obtenido de Tecnologias-informacion.com: <https://www.tecnologias-informacion.com/sigerencial.html>
- Sistemas de Procesamiento de Transacciones | Tecnologias-informacion. (s.f.). *Tecnologias-informacion.com*. Obtenido de Tecnologias-informacion.com: <https://www.tecnologias-informacion.com/procesamientotransacciones.html>
- Sistemas de Soporte a la Decisión (DSS) | Sinnexus.com. (s.f.). *Sinnexus.com*. Obtenido de Sinnexus.com: https://www.sinnexus.com/business_intelligence/sistemas_soporte_decisiones.aspx
- Te explicamos en qué consiste y cómo realizar la Factibilidad Económica | {Tomas}. (2022). *Tesis y Másters Chile*. Obtenido de Tesis y Másters Chile: <https://tesisymasters.cl/factibilidad-economica/>
- Técnicas de la metodología cualitativa | López, N. L. (s.f.). *Unidades de Apoyo para el Aprendizaje - CUAIEED - UNAM*. Obtenido de Unidades de Apoyo para el Aprendizaje - CUAIEED - UNAM: https://repositorio-uapa.cuaieed.unam.mx/repositorio/moodle/pluginfile.php/2558/mod_resource/content/1/UAP-A-Tecnicas-Metodologia-Cualitativa/index.html
- Tipos de Factibilidad | My first document. (s.f.). *Issuu*. Obtenido de Issuu: https://tomi.digital/es/164314/trabajo-humano-de-la-empresa?utm_source=google&utm_medium=seo
- Tipos de investigación cuantitativa: Cuáles son y ejemplos | Narvaez, M. (2019). *QuestionPro*. Obtenido de QuestionPro: <https://www.questionpro.com/blog/es/tipos-de-investigacion-cuantitativa/>
- Tipos de variables independientes y dependiente | Tesisdeceroa100.com. (s.f.). *Tesisdeceroa100.com*. Obtenido de Tesisdeceroa100.com: <https://tesisdeceroa100.com/lo-que-nadie-te-conto-sobre-tipos-de-variables-en-investigacion/>
- Tipos y niveles de lenguajes de programación ejemplos | AAU. (2024). *Tipos y niveles de lenguajes de programación ejemplos 2024*. Obtenido de Tipos y niveles de lenguajes de programación ejemplos 2024: <https://www.aauuniv.com/s/blog/lenguajes-de-programacion/>
- Trazabilidad de Activos | Tech-ID. (2023). *Tech-ID*. Obtenido de Tech-ID: <https://www.tech-id.com/procesos/trazabilidad-activos/>
- unir. (2024). *unir*. Obtenido de <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>
- Vargas Rodríguez. (2022). *Sibdi UCR*. Obtenido de Sibdi UCR: <https://repositorio.sibdi.ucr.ac.cr/items/80fed87a-fd00-4e71-92fd-f4b0b38b9baf>

Variables en la Investigación | Stewart, L. (2025). *ATLAS.ti*. Obtenido de ATLAS.ti:
<https://atlasti.com/es/research-hub/variables-investigacion>

ZENDESK. (2025). <https://www.zendesk.com.mx/blog/sistema-de-seguimiento-de-tickets/>. Obtenido de
<https://www.zendesk.com.mx/blog/sistema-de-seguimiento-de-tickets/>.