

MEMORIA ESCRITA DEL PROYECTO

CFGS Administración de Sistemas Informáticos en Red

Agentes IA y Automatizaciones: ¿Complemento, o competencia de los trabajadores del sector IT?

Autor: Miqueas Molina Delgado

Tutor: Laura Bermúdez Galbarriatu

Fecha de entrega: 23/04/2025

Convocatoria: 2s2425

Documentos del proyecto:  Proyecto final ASIR Ilerna carpeta compartida



Índice

1. Introducción	2
1.1. Motivación	3
1.2. Abstract	4
1.3. Objetivos propuestos (generales y específicos)	5
2. Estado del arte	6
2.1. Impacto general de la IA y automatización en el sector IT	6
2.2. Agentes de IA en el entorno IT	7
2.3. Automatización robótica de procesos (RPA) en operaciones IT	8
2.4. Plataformas de integración y automatización	9
2.5. Identificación de vacíos, riesgos y oportunidades	10
3. Metodología usada	10
3.1. Fases de la metodología	11
3.2. Limitaciones del estudio	12
4. Tecnologías y herramientas utilizadas en el proyecto	13
4.1. Tecnologías y herramientas analizadas	14
4.2. Herramientas utilizadas para la realización del proyecto	15
5. Planificación, diagnóstico y contexto laboral	16
5.1. Planificación	16
5.2. Diagnóstico	18
5.3. Contexto laboral	19
6. Análisis del proyecto	20
6.1. Impacto funcional y beneficios en las áreas de trabajo ASIR	20
6.2. Riesgos, limitaciones y consideraciones de ciberseguridad	23
6.3. Requisitos técnicos para la integración	25
6.4. Sostenibilidad, escalabilidad y viabilidad de la solución	27
6.5. Síntesis del análisis y preparación para el diseño	30
7. Diseño del proyecto	31
7.1. Arquitectura general de la solución	32
7.2. Flujos de trabajo automatizados propuestos	34
7.3. Seguridad, gobierno y mantenimiento del sistema	37
7.4. Plan de formación para los técnicos ASIR	38
8. Despliegue y pruebas	40
8.1. Plan de despliegue teórico	40
8.2. Plan de pruebas y validación	44
8.3. Criterios de éxito y aceptación	47
9. Conclusiones	48
10. Vías futuras	50
11. Bibliografía/Webgrafía	53
12. Glosario	55

1. Introducción

El sector tecnológico siempre ha sido uno de los ámbitos con una evolución más rápida, pero la llegada de las **automatizaciones**, y en los últimos años de la **Inteligencia Artificial** (en adelante IA), han hecho que esta evolución tenga un ritmo vertiginoso. Por un lado, estas herramientas prometen mejorar la eficiencia y asumir tareas rutinarias; por otro, alimentan el temor de que puedan desplazar a profesionales humanos. Organismos internacionales estiman que la IA afectará a casi el 40% de los empleos mundiales (IMF, 2024), reemplazando algunos y complementando otros. Aunque estos avances están teniendo repercusión en todos los sectores económicos y profesionales, este proyecto se centrará en el sector de las tecnologías de la información (IT/TI), que nos atañe directamente.

El panorama actual no tiene precedentes, se están redefiniendo las formas en las cuales se realizan las tareas, y los roles laborales. Hay muchos roles que hasta hace muy poco no existían, o para los cuales había muy poca demanda, y, por otro lado, otros que están cayendo en desuso. El sector tecnológico es, en parte, el epicentro de esta revolución, siendo un impulsor, pero al mismo modo un receptor de sus efectos.

Los trabajadores que se dedican a la **Administración de Sistemas Informáticos en Red** (en adelante ASIR), son los responsables de mantener la infraestructura tecnológica de las organizaciones, un trabajo vital. Se enfrentan a un nuevo escenario, donde nuevas herramientas, agentes inteligentes potenciados con IA, plataformas de **automatización robótica de procesos** (en adelante RPA), prometen optimizar y complementar ciertos aspectos de su trabajo diario. Esta es la cara bonita de la inteligencia artificial y la automatización, unos trabajadores con más tiempo libre y con menos tareas repetitivas. Sin embargo, estas herramientas podrían llegar a ser competidores o poner en peligro muchos puestos de trabajo también.

Es precisamente esa dualidad la que da título al proyecto: ¿Son estas herramientas un **complemento**, que potencie las capacidades del trabajador IT, o representan una **competencia** directa que amenaza su estabilidad laboral? Comprender la situación actual y el posible futuro (aunque no tengamos una bola de cristal) es clave para que tanto los profesionales actuales, como los futuros (estudiantes actuales de ASIR) puedan surfear

esta transición, identificando oportunidades y adquiriendo las habilidades necesarias para seguir siendo valiosos y relevantes en el mercado laboral.

El presente trabajo de investigación se adentra en dicha cuestión, analizando de forma estructurada distintas categorías de herramientas. No se trata de un proyecto de simulación o implementación práctica, sino de investigación aplicada al campo de la administración de sistemas: se considerarán fuentes actuales para reflejar el rápido avance de esta temática. Quedan fuera de este estudio aspectos no directamente relacionados con el sector IT (por ejemplo, impacto de la IA en otros sectores) salvo cuando sean útiles para analogías o contexto.

Comenzaremos desarrollando el **estado del arte** actual de la aplicación de Agentes IA y automatizaciones en tareas IT. Posteriormente, realizaremos un **Análisis** más detallado de herramientas específicas, clasificadas en 3 grupos, Agentes IA, soluciones de RPA Empresarial y plataformas de Integración o Automatización generalistas. Estas se complementan con las estudiadas en el temario de ASIR (Bash scripting, PowerShell, SQL, Zabbix y herramientas IDE como Visual Studio Code). Se evaluará su funcionalidad, aplicabilidad en distintos roles IT y su potencial impacto como complemento o competencia. Finalmente, se presentarán las **Conclusiones** de la investigación y se esbozarán posibles **Vías futuras** y posibles mejoras.

1.1. Motivación

La elección de este tema de investigación para el proyecto surge de una combinación de intereses académicos, profesionales y personales. En primer lugar, como estudiante del ciclo formativo de grado superior en ASIR, es necesario, tanto **a nivel académico como profesional**, que entendamos las herramientas y tendencias que están modelando e influyendo en nuestro sector, el sector IT. La IA con sus agentes o asistentes (GitHub Copilot, Cursor) y la automatización, incluyendo herramientas para realizar RPA (UiPath, Automation Anywhere, PowerAutomate), así como las plataformas de integración tipo Make, Zapier y n8n, son tecnologías que los futuros técnicos tendrán que conocer y manejar. Es imprescindible mantenerse actualizado y competitivo.

En segundo lugar, **a nivel personal**, tengo un profundo interés en entender cómo va evolucionando la tecnología, y como la IA y las automatizaciones están redefiniendo o potenciando las capacidades del humano y los entornos laborales. Investigar si estas

herramientas van a ser mis colaboradores o mis competidores, en el ámbito IT, no solo satisface mi curiosidad intelectual, sino que también me permite obtener una perspectiva informada sobre cómo podré, como profesional, adaptarme y aprovechar esta tecnología en mi beneficio.

Para terminar, considero que la importancia que estas herramientas tienen y van a tener en la industria tecnológica es innegable. Las empresas invierten cada vez más en estas soluciones, intentando optimizar sus procesos, mejorar su eficiencia y obtener ventajas competitivas. Debido a eso, analizar este fenómeno desde la perspectiva del trabajador IT nos puede proporcionar una visión valiosa sobre qué habilidades serán demandadas y las posibles trayectorias que como profesionales podemos tener dentro del sector.

1.2. Abstract

The current technological landscape is undergoing rapid transformation, especially with the rise of **automation tools and artificial intelligence** (AI). The Information Technology (IT) sector, particularly roles related to Computer Systems Administration (ASIR), is directly affected by the emergence of AI agents, Robotic Process Automation (RPA), and integration platforms.

This project investigates whether these technologies act as **complements** that enhance human capabilities or as **competitors** threatening job stability. This research examines this duality by analyzing specific, accessible tools (available in free trials, students, or community editions).

The study includes a State-of-the-Art review of AI and automation within IT tasks, followed by a comparative analysis of selected tools across three categories: **AI Agents** (GitHub Copilot, Cursor), **Enterprise RPA** (UiPath, Automation Anywhere, Power Automate) and **Integration/Automation Platforms** (Make, Zapier, n8n). These tools complement the ones studied in our IT degree curriculum (Bash scripting, PowerShell, SQL, Zabbix, Visual Studio Code). The analysis evaluates their features and applications to different IT domains (sysadmin, networking, development, etc.) and discusses their potential impact, either as complementary assets or competitive forces.

The primary aim is to provide a clear overview and actionable insights for both current and future IT professionals, helping them navigate this transition, by identifying key skills required in this evolving ecosystem.

1.3. Objetivos propuestos (generales y específicos)

Objetivo general:

1. **Evaluar el impacto** de los agentes IA (asistentes) y las herramientas de automatización (RPA e integración) en diversos roles profesionales del sector IT, determinando si actúan como **complementos** que potencian sus habilidades y capacidades, o como **competidores** que pudieran llegar a amenazar su estabilidad laboral.

Objetivos específicos:

1. **Analizar** el estado del arte:
 - a. Investigar y documentar las principales tendencias y avances en el uso de agentes IA y soluciones de automatización, especialmente en el ámbito IT.
 - b. Revisar estudios, casos de uso y experiencias que aborden tanto la aplicación de RPA empresarial, como las plataformas de integración y automatización.
2. **Clasificar y comparar** herramientas:
 - a. Analizar distintas herramientas, clasificadas en tres categorías: agentes inteligentes (GitHub Copilot, Cursor), RPA empresarial (UiPath, Automation Anywhere, Power Automate) y soluciones de automatización e integración generalistas (Make, Zapier, n8n).
 - b. Evaluar sus funcionalidades, ventajas, limitaciones y su interacción con las herramientas del temario de ASIR (Bash scripting, PowerShell, SQL, Zabbix, Visual Studio Code).
3. **Evaluar** el impacto en los roles IT:
 - a. Analizar cómo estas herramientas y tecnologías, influyen en las tareas y competencias de los profesionales IT, en especial de los Administradores de Sistemas Informáticos en Red (ASIR).
 - b. Investigar el posible impacto en la eficiencia, coste y calidad de los procesos, así como la implicación en la formación y actualización profesional necesaria.
4. **Determinar** implicaciones laborales y de formación:

- a. Identificar competencias y habilidades que serán esencial en el futuro mercado IT, a partir de la integración de estas herramientas.
 - b. Proponer estrategias y recomendaciones, tanto para los trabajadores actuales como los futuros profesionales de ASIR, para que puedan adaptarse a estos cambios sin perder relevancia en el mercado laboral.
5. **Proponer** futuras líneas de investigación y mejora:
- a. Documentar los objetivos inicialmente planteados en la petición de tema que, por limitaciones de alcance o tiempo, no han podido incluirse en el proyecto.
 - b. Esbozar posibles vías futuras de investigación para profundizar en el estudio de la integración de la IA y la automatización en el entorno laboral IT.
 - c. Sugerir mejoras en la implementación y adopción de estas tecnologías, para maximizar su potencial como herramientas complementarias.

2. Estado del arte

Esta sección presenta una revisión exhaustiva de la literatura y del conocimiento actual sobre el **impacto de la IA y las herramientas de automatización** en el sector IT, con un enfoque particular en los roles y tareas de los técnicos de ASIR. El objetivo es poner en contexto la investigación, identificar las tendencias y herramientas más relevantes para detectar como estas tecnologías están transformando el panorama laboral en IT. Esta revisión servirá como base para un posterior análisis de herramientas específicas y para considerar su papel como complemento o competencia para los trabajadores IT.

2.1. Impacto general de la IA y automatización en el sector IT

Diversos estudios e informes de distintas consultoras internacionales, han destacado como la **adopción de la IA y la automatización** está transformando el panorama del sector IT, redefiniendo procesos operativos, roles profesionales y estrategias empresariales.

Un informe de McKinsey & Company (2023) estima que las tecnologías de IA, incluyendo la generativa, podrían automatizar entre el 60% y el 70% de las tareas actuales de los trabajadores IT para 2030 (algunas de forma total, otras de forma parcial), con un impacto significativo en áreas como gestión de datos, soporte técnico y administración de

sistemas (McKinsey & Company, 2023). Esta proyección no implica per se una eliminación total de los empleos, sino una reorientación hacia actividades menos repetitivas, de mayor valor, como las más creativas o estratégicas, lo que refuerza la pregunta central del presente proyecto: ¿complementan o compiten estas tecnologías con los profesionales IT?

2.2. Agentes de IA en el entorno IT

Con respecto a los agentes de IA, Gartner (2024) identifica a la **IA agéntica** (sistemas que planifican y ejecutan tareas con cierto grado de autonomía), como una tendencia clave para 2025 (ver Figura 1), (Gartner, 2024). Esta tendencia promete mejoras en eficiencia, pero supone un desafío también en términos de capacitación. Los técnicos ASIR, responsables de mantener y manejar sistemas críticos, podrían beneficiarse de estas tecnologías para reducir cargas administrativas.

Figura 1.

Principales tendencias tecnológicas para 2025, según Gartner.



Nota: Adaptado de *Top 10 strategic technology trends for 2025*, por Gartner, 2024. (<https://www.gartner.com/en/articles/top-technology-trends-2025>). © 2024 Gartner, Inc. y/o sus afiliados. Todos los derechos reservados.

Estos **agentes de IA**, están redefiniendo la productividad y los flujos de trabajo en el sector IT. Entre ellos se incluyen herramientas de asistencia (**asistentes IA**, los que analizamos en el proyecto) como GitHub Copilot y Cursor, diseñadas para apoyar a los profesionales en la programación y gestión de sistemas, así como agentes autónomos u operadores, que ejecutan procesos más complejos, con una intervención humana mínima. Según un estudio de GitHub, GitHub Copilot incrementa la productividad de los desarrolladores hasta un 55%, al sugerir código y automatizar tareas repetitivas, como la estructura de los scripts o la depuración (GitHub, 2023). Esto es destacable, para los técnicos ASIR, que a menudo gestionan configuraciones de servidores mediante scripts de automatización.

Por otro lado, Cursor, es un entorno de desarrollo impulsado por IA, que permite generar código a partir de instrucciones en lenguaje natural, reduciendo el tiempo de desarrollo hasta en un 30%, según ciertos artículos (Daily.dev, 2024). Estas herramientas optimizan procesos, pero también podrían generar una dependencia excesiva y limitar el aprendizaje profundo de conceptos técnicos, un riesgo para los estudiantes y futuros trabajadores de ASIR.

Por último, los **agentes autónomos u operadores**, como AutoGPT, están emergiendo en IT para llevar a cabo tareas como monitorización de redes o análisis predictivo. Gartner (2024) indica que estos agentes serán claves en el 2025, proyectando que el 15% de las decisiones laborales diarias en IT se tomarán de forma autónoma por sistemas de IA. Algunos ejemplos son la detección de fallos en infraestructura, una tarea común para los trabajadores en ASIR. Sin embargo, su implementación requiere supervisión humana para prevenir errores críticos (Gartner, 2024).

2.3. Automatización robótica de procesos (RPA) en operaciones IT

La **automatización robótica de procesos** (RPA) se ha consolidado como una tecnología relevante para optimizar ciertas operaciones dentro de los departamentos de IT (ITOps, operaciones IT). A diferencia de la automatización tradicional, que se basaba en scripts o APIs más complejas, RPA utiliza “robots” de software para imitar comportamientos humanos y automatizar tareas repetitivas basadas en reglas. Estos bots interactúan con aplicaciones y sistemas de la misma manera en la que lo haría un usuario humano, pero con mayor velocidad y precisión (Automation Anywhere, s.f.).

En el contexto de los técnicos ASIR, la RPA puede aplicarse en tareas distintas, como la gestión de tickets e incidencias, la monitorización de sistemas, la generación de informes y la administración de usuarios. Liberando tiempo de estas tareas, los profesionales pueden enfocarse en tareas de mayor valor, como la mejora de la infraestructura o la seguridad de los sistemas (SAP, s.f.). Entre las plataformas líderes en RPA, destacan UiPath, Automation Anywhere, Blue Prism y Microsoft Power Automate. Estas herramientas ofrecen soluciones robustas y escalables. (Valtx, 2024).

Además, la implementación de RPA en entornos IT contribuye a reducir errores humanos, mejora la eficiencia operativa y garantiza una mayor consistencia en los procesos. Por lo tanto, parece necesario que los técnicos ASIR adquieran competencias en el diseño, implementación y supervisión de estas automatizaciones (CyberArk, s.f.).

2.4. Plataformas de integración y automatización

Junto a RPA, otra categoría importante de herramientas de automatización en el entorno IT, son las **plataformas de integración como servicio** (iPaaS) y las plataformas de automatización **low-code/no-code** (en adelante LCNC), como pueden ser Make (anteriormente Integromat), Zapier y n8n. Estas plataformas, permiten conectar aplicaciones y automatizar procesos sin necesidad de saber programación avanzada, siendo útiles para optimizar operaciones IT.

Zapier, por ejemplo, facilita la integración de herramientas como gestión de tickets, como ServiceNow, con plataformas de comunicación, como Slack, automatizando notificaciones de incidencias. Según Zapier (s.f.), estas plataformas pueden ahorrar varias horas semanales al automatizar tareas repetitivas, beneficiando a los técnicos ASIR de una reducción de carga administrativa (Zapier, s.f.).

Make por su parte, ofrece flujos de trabajo más complejos, como la sincronización de datos entre servidores y aplicaciones en la nube, una tarea que podría ser común para los trabajadores ASIR, gracias a su capacidad para conectar más de 1000 aplicaciones (Leadpicker, s.f.).

Por último, n8n, permite automatizar alertas de incidencias, restablecimiento de contraseñas, y tiene multitud de plantillas predefinidas para trabajadores ITOps, así como más de 400 integraciones, mejorando la estabilidad de la infraestructura y reduciendo errores humanos (n8n, s.f.).

2.5. Identificación de vacíos, riesgos y oportunidades

La revisión realizada hasta el momento, del estado del arte, en las secciones 2.1 a 2.4, evidencia cómo la IA, la RPA y las plataformas de integración están transformando el sector IT, impactando en roles como los de los técnicos ASIR. Todas estas herramientas, agentes operadores y asistentes IA, plataformas de RPA y como soluciones de integración, buscan mejorar la productividad y eliminar las tareas repetitivas.

Sin embargo, existen vacíos significativos, como la **falta de investigación** sobre cómo estas tecnologías afectan las competencias y la empleabilidad de los técnicos ASIR, especialmente en soporte técnico y administración de sistemas y cómo interactúan con herramientas tradicionales, como Bash scripting, PowerShell o Zabbix. Es un riesgo potencial, que estas herramientas, al automatizar tareas clave de ASIR e ITOps, como la gestión de incidencias o el soporte básico, puedan competir directamente con estos profesionales, reduciendo la demanda de roles operativos y desplazándolos si no se adaptan al cambio.

Además, no se abordan **estrategias para mitigar riesgos** como la dependencia excesiva de las automatizaciones o robots de software, o los desafíos de configuración técnica de los mismos, lo que resalta la necesidad de formación actualizada.

Estas lagunas, riesgos y oportunidades, justifican un análisis más profundo, que será realizado en la sección 6, Análisis del proyecto, donde se evaluará cómo estas herramientas pueden integrarse en los flujos de trabajo de ASIR, y si se puede equilibrar eficiencia y desarrollo personal para garantizar una adaptación sin riesgos a un entorno laboral automatizado. Con base en esta revisión del estado del arte, la siguiente sección detalla la metodología empleada para analizar estas herramientas y su impacto en los técnicos ASIR.

3. Metodología usada

En esta sección se describe la metodología usada para el presente proyecto de investigación teórica. Dado que no se ha desarrollado un proyecto de implementación, se excluyen metodologías más orientadas a la simulación o pruebas prácticas. El objetivo es analizar el impacto de la IA y la automatización en roles de ASIR mediante una revisión

bibliográfica y posterior análisis comparativo, respondiendo a la pregunta central: ¿son un complemento, o una competencia?

Este proceso se estructura en varias fases, detalladas a continuación, que permiten recopilar información relevante, seleccionar herramientas adecuadas y evaluar su impacto en los perfiles profesionales del sector IT.

3.1. Fases de la metodología

3.1.1. Técnicas de recolección de datos

La información utilizada en este proyecto ha sido recopilada a través de fuentes secundarias fiables y actualizadas. La investigación comenzó con la elaboración del estado del arte (sección 2), llevada a cabo mediante una revisión bibliográfica exhaustiva. Se recopilaron datos de **fuentes secundarias recientes**, usando buscadores tanto generales como académicos (Google, Google Scholar) utilizando distintas palabras clave, tanto en inglés como en español, como “AI Agents in 2025” (para tendencias de IA), “AI and ITOps” (para operaciones IT), “automatización en ASIR” (para roles específicos), “RPA empresarial” y “plataformas de integración”.

Además, se utilizaron **informes de consultoras reconocidas**, como McKinsey & Company y Gartner, blogs especializados, como GitHub, y sitios oficiales de distintas herramientas, como Zapier y n8n. Las fuentes se seleccionaron priorizando su fiabilidad, actualidad y relevancia para el sector IT, con un enfoque específico en las tareas comunes de los técnicos ASIR, como gestión de sistemas, soporte técnico y uso de scripts (scripting).

3.1.2. Criterios de selección de herramientas

La segunda fase consistió en **identificar las herramientas** a analizar, basándose en dos criterios principales: **accesibilidad y relevancia** para las tareas de ASIR. Se priorizaron herramientas aplicables a actividades como scripting, gestión de tickets, monitorización de sistemas e integración de aplicaciones, que estuvieran disponibles en versiones gratuitas, de prueba, con versión comunidad o estudiante (student) o con documentación pública accesible para un proyecto académico como este.

Esto llevó a la selección de herramientas en **tres categorías distintas**: agentes asistentes IA (GitHub Copilot y Cursor), soluciones de RPA empresarial (UiPath, Automation

Anywhere (en adelante AA) y Power Automate (en adelante PA)) y plataformas de integración LCNC (Make, Zapier y n8n). Dentro de la categoría de agentes IA, se priorizó el análisis detallado de herramientas tipo asistentes IA, dada su mayor accesibilidad, frente a agentes operadores, más autónomos, cuya evaluación presentaba mayores barreras de coste y complejidad técnica.

Estos criterios hicieron posible que las herramientas fueran tanto representativas de los distintos niveles de automatización, como viables para el análisis teórico posterior. Estas herramientas sirven como complemento a las propias estudiadas en el ciclo formativo de grado superior ASIR (Bash scripting, PowerShell, SQL, Zabbix, y herramientas IDE como Visual Studio Code).

3.1.3. Evaluación comparativa

La tercera fase metodológica consistió en una evaluación comparativa, no técnica, de las herramientas seleccionadas, combinada con un análisis de su impacto laboral en los técnicos ASIR. La evaluación se basó en distintos **criterios** como: funciones clave y nivel de automatización (qué tipo de tareas resuelven y qué nivel de intervención humana necesitan), aplicabilidad al entorno ASIR (en qué medida pueden complementar, mejorar o sustituir las tareas propias del profesional técnico), facilidad de adopción (curva de aprendizaje, interfaz, necesidad de conocimientos técnicos) e impacto potencial en el empleo y la formación (qué habilidades nuevas se requieren y qué tareas se podrían ver desplazadas o potenciadas). También se analizó cómo estas herramientas interactúan con las del temario de ASIR.

Además, se buscaron **indicadores** en distintos estudios y encuestas (productividad, ahorro de tiempo y/o costes, percepción del usuario), para obtener una perspectiva más clara de su impacto real.

3.2. Limitaciones del estudio

El enfoque teórico adoptado en este proyecto permite un análisis amplio y genérico del impacto de la IA y la automatización en los técnicos ASIR, pero presenta también ciertas limitaciones, inherentes a este tipo de trabajos de investigación.

En primer lugar, la **ausencia de pruebas empíricas**. Las conclusiones se basan en datos teóricos, y no en implementaciones reales. No se ha realizado un entorno de pruebas que permita contrastar el impacto de estas herramientas de forma directa. Las conclusiones dependerán de la calidad y objetividad de la información a la que se tenga acceso. En segundo lugar, la **selección de las herramientas**, limitada por criterios de relevancia y accesibilidad, deja de lado otras muchas soluciones que existen en el mercado, tanto de IA (agentes operadores), como de RPA (Blue Prism) e integración (Gumloop), por mencionar algunos ejemplos, que no han podido incluirse en el análisis detallado.

Finalmente, la **velocidad vertiginosa** a la que la IA y la automatización evolucionan, hace que cada día, cada semana, cada mes, aparezcan herramientas nuevas, con casos de uso distintos. Es probable que nuevas funcionalidades, variantes y estudios surjan rápidamente, pudiendo matizar o actualizar algunas de las conclusiones aquí presentadas. Todas estas limitaciones serán consideradas de cara a las propuestas de mejora y vías futuras que se presentan en la sección 10 del proyecto.

4. Tecnologías y herramientas utilizadas en el proyecto

Este apartado presenta las herramientas analizadas para el análisis (tanto las propias del temario ASIR como las externas), agrupadas en 3 categorías: agentes asistentes IA, soluciones de RPA empresarial y plataformas de integración low-code/no code (LCNC). Estas herramientas se seleccionaron siguiendo los criterios de accesibilidad y relevancia definidos en la metodología (sección 3.1.2).

A continuación, se describen sus características principales y propósito general, con un enfoque en su aplicabilidad en las tareas de los técnicos ASIR, como scripting, gestión de sistemas y redes, monitorización o integración de flujos operativos. También se explora como estas herramientas complementan las estudiadas en el temario ASIR, proporcionando un marco para el análisis comparativo que se desarrollará en la sección 6.

4.1. Tecnologías y herramientas analizadas

4.1.1. Agentes asistentes IA

Los agentes asistentes IA analizados son **GitHub Copilot y Cursor**. Estas herramientas complementan las del temario de ASIR al integrarse en Visual Studio Code mediante extensiones, mejorando la productividad al generar código que complementa scripts escritos manualmente en Bash o PowerShell. El primero de ellos, GitHub Copilot, se trata de un asistente de programación basado en IA que sugiere fragmentos de código, funciones completas o comandos en la terminal, facilitando el trabajo tanto en entornos de scripting como Bash y PowerShell, como en lenguajes más complejos. En el contexto de ASIR, permite agilizar tareas como la automatización de scripts (por ejemplo, proponer un script para automatizar copias de seguridad en un servidor Linux), la escritura de comandos y la depuración de errores.

Del mismo modo, Cursor permite crear scripts en PowerShell a partir de instrucciones en lenguaje natural (ej., "generar un script para gestionar usuarios en Active Directory"), facilitando tareas administrativas en entornos Windows. Es un entorno de desarrollo potenciado por IA que permite generar, editar y probar código de forma contextual, con una interfaz que integra funciones de copiloto y chat. Para el entorno de ASIR resulta útil en tareas de programación de scripts y automatización de procesos, sustituyendo o ampliando el uso tradicional de IDEs como Visual Studio Code.

4.1.2. Soluciones de RPA empresarial

Las soluciones de RPA empresarial estudiadas incluyen **UiPath, Automation Anywhere y Power Automate**. Estas herramientas complementan las del temario de ASIR de diversas formas. Power Automate puede integrarse con PowerShell para automatizar tareas complejas, como la gestión masiva de permisos en carpetas compartidas, ejecutando scripts de PowerShell de forma desatendida. Además, se integra fácilmente con sistemas Windows y servicios como Outlook, SharePoint o Excel. En el contexto ASIR, permite automatizar tareas administrativas, enviar alertas automáticas o procesar eventos en sistemas monitoreados.

Por su parte, UiPath puede ejecutar consultas SQL automáticamente para generar informes de bases de datos, como estadísticas de uso de recursos en una empresa, reduciendo la intervención manual del técnico ASIR. Además, interactúan con Zabbix al recibir alertas de monitorización y automatizar respuestas, como reiniciar un servidor tras una alerta de fallo o escalar incidencias a un sistema de tickets.

4.1.3. Plataformas de integración LCNC

Las plataformas de integración low-code/no-code (LCNC) seleccionadas son **Make**, **Zapier** y **n8n**. Estas plataformas complementan las herramientas del temario de ASIR al facilitar la automatización de flujos operativos. Permiten construir flujos de integración entre servicios web. Son muy útiles para automatizar notificaciones, sincronización de datos o tareas relacionadas con tickets y correo electrónico. Por ejemplo, se integran con Zabbix mediante su API para automatizar tareas de monitorización.

Citemos el caso de n8n, plataforma de código abierto con enfoque más técnico que permite crear flujos complejos, integrando scripts personalizados y servicios de terceros. Resulta especialmente útil para tareas como restablecimientos automáticos de contraseñas o integraciones con bases de datos, combinando SQL con flujos visuales. Además, n8n cuenta con un nodo dedicado para Zabbix, puede recuperar alertas (ej., un servidor caído) y enviar notificaciones a Slack o Microsoft Teams, o incluso crear ítems automáticamente para monitorizar nuevos parámetros (n8n, s.f.).

Por último, estas plataformas pueden integrarse con repositorios como GitHub para automatizar despliegues tras un commit (una acción que guarda los cambios realizados en el código dentro de un repositorio como GitHub) realizado desde entornos como Visual Studio Code. Además, en el caso de n8n, es posible ejecutar scripts predefinidos (por ejemplo, en PowerShell) en servidores configurados, lo que permite automatizar tareas como el restablecimiento masivo de contraseñas en respuesta a eventos específicos.

4.2. Herramientas utilizadas para la realización del proyecto

Dado que este proyecto es de naturaleza teórica y no incluye implementación práctica, las herramientas utilizadas para su realización fueron principalmente de investigación, documentación y presentación.

Se emplearon buscadores como Google y Google Scholar para recopilar información, utilizando palabras clave en español e inglés (detalladas en la Sección 3.1.1), además de consultar sitios oficiales de las herramientas analizadas (por ejemplo, GitHub, Zapier, n8n) para obtener información técnica y casos de uso. Para la gestión de referencias y la redacción del documento, se utilizaron herramientas de procesamiento de texto como **Microsoft Word y Google Docs**, que permitieron organizar el contenido, estructurar las secciones y mantener un formato consistente. También se utilizó **Google Drive** para el almacenamiento y la organización colaborativa de archivos.

Para la elaboración de las diapositivas explicativas del proyecto, se utilizó la herramienta **Gamma App**, que permite generar presentaciones visuales e interactivas de forma eficiente, adecuada para explicar conceptos técnicos de forma clara. Por último, para la grabación del vídeo explicativo del proyecto, se utilizó **OBS Studio**, una herramienta de código abierto (open source) ampliamente utilizada para grabar presentaciones y capturar pantalla. No se requirieron otras herramientas adicionales, ya que no se realizaron pruebas prácticas ni simulaciones funcionales de automatización.

A continuación, se analizará el contexto en el que estas herramientas podrían implementarse, así como el impacto previsto en el entorno laboral del técnico ASIR.

5. Planificación, diagnóstico y contexto laboral

Este apartado presenta la planificación temporal del proyecto, un diagnóstico del problema concreto que se pretende abordar, dentro del ámbito de la Administración de Sistemas Informáticos en Red (ASIR), y en que contexto laboral se sitúa esta investigación. Se ha seguido una planificación semanal desde el inicio del proyecto a principios de marzo, hasta su entrega final, a finales de abril, combinando tareas de investigación, análisis, redacción y presentación. Además, se analiza el diagnóstico del problema que motiva el estudio y el entorno laboral donde estas tecnologías podrían aplicarse con mayor impacto.

5.1. Planificación

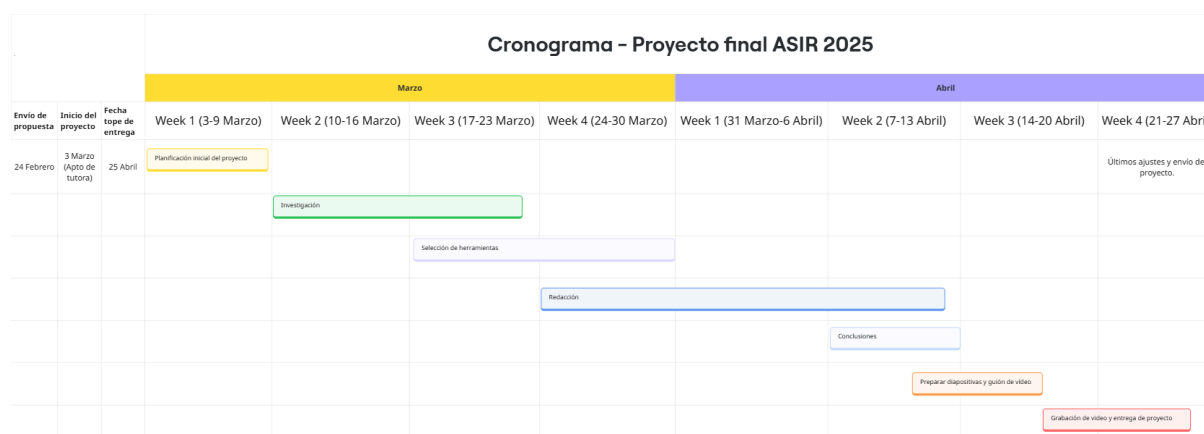
La planificación del proyecto se estructuró para cumplir con los objetivos definidos en la sección 1.3, comenzando con el envío de la propuesta de proyecto el lunes 24 de

febrero de 2025 y terminando con la entrega del proyecto el 23 de abril de 2025, unos dos meses de trabajo. A continuación, se presenta una descripción textual acompañada al **diagrama de Gantt**, detallando las etapas, fechas y recursos utilizados. El proyecto se organiza en siete diferentes fases:

1. Planificación inicial del proyecto (en amarillo) – Se definieron los objetivos del proyecto, la estructura de las secciones, y se identificaron los recursos necesarios. También se realizó una revisión inicial de fuentes para el estado del arte.
2. Investigación (en verde) – Se recopiló información para las secciones 2 (estado del arte) y 3 (metodología).
3. Selección de herramientas (en morado) – Se definieron los criterios para elegir las herramientas y seleccionaron las herramientas mismas.
4. Redacción (en azul oscuro) – Se desarrollaron las diferentes secciones del documento, haciendo especial hincapié en el análisis del proyecto, la sección más larga de este proyecto de investigación.
5. Conclusiones (en azul claro) – Se redactaron las secciones 9 y 10 (conclusiones, vías futuras), sintetizando los hallazgos del proyecto, limitaciones y oportunidades.
6. Preparación de diapositivas y guion de video (en naranja) – Se elaboraron las diapositivas para la presentación y se escribió el guion para el vídeo explicativo.
7. Grabación de video y entrega del proyecto – Se realizó la grabación del vídeo explicativo, capturando las diapositivas y explicaciones narradas, se preparó el material para la entrega y se formalizó dicha entrega en la plataforma.

Figura 2.

Diagrama de Gantt que describe la duración de las fases del proyecto.



Un desafío significativo fue la falta de estudios específicos sobre el impacto de estas herramientas en roles ASIR, lo que requirió un análisis interpretativo. Además, el tiempo limitado impidió analizar herramientas más avanzadas, como agentes operadores IA, lo que se propone como vía futura en la sección 10.

Este cronograma permitió una ejecución ordenada del proyecto, asegurando que todas las etapas se completaran dentro del plazo establecido. Cada fase fue diseñada para tener una duración realista, permitiendo el solapamiento de tareas donde fuera necesario (tal y como se ve en el cronograma). En términos de cumplimiento, **la mayoría de las tareas** han seguido el calendario previsto. Si es cierto que cuando se hizo la estimación inicial, se calculó una dedicación similar de horas cada semana, y finalmente estas fueron distintas, teniendo que trabajar más de lo previsto en alguna de ellas (semanas desde el 4 de marzo al 3 de abril).

5.2. Diagnóstico

El proyecto parte de una situación común en el ámbito de la Administración de Sistemas Informáticos en Red (ASIR): la sobrecarga operativa de los técnicos debido a la gestión manual de tareas repetitivas, como la monitorización de infraestructura y la resolución de incidencias básicas.

En muchas organizaciones, los técnicos ASIR dedican una parte significativa de su tiempo a actividades como la gestión de tickets de soporte (ej., restablecimiento de contraseñas), la generación de informes SQL para estadísticas de uso de recursos, y la monitorización manual de alertas en herramientas como Zabbix. Esto genera demoras en la respuesta a incidencias críticas y limita su capacidad para realizar tareas estratégicas, como la optimización de la infraestructura o la implementación de medidas de ciberseguridad. Este problema se agrava en entornos con recursos limitados, donde los técnicos deben manejar múltiples sistemas operativos (Linux y Windows) y bases de datos simultáneamente, lo que incrementa el riesgo de errores humanos y reduce la eficiencia operativa.

Con la irrupción y adopción de la IA y las automatizaciones, podríamos pensar que esa carga de trabajo se está viendo, y se verá aliviada. Sin embargo, nos encontramos con la falta de claridad sobre cómo estas herramientas impactan en las funciones, competencias y empleabilidad de los técnicos de ASIR. Muchos estudiantes y profesionales en este

campo desconocen si estos avances representan una oportunidad para mejorar su productividad, o una amenaza que podría reemplazar ciertas tareas que históricamente han sido parte esencial de su rol. Este es el **problema específico** que el proyecto busca analizar. Si estas herramientas pueden ser un complemento efectivo que ayude a los técnicos ASIR en su labor diaria, o más bien una competencia que amenace su estabilidad laboral.

5.3. Contexto laboral

El contexto laboral en el que se enmarca este estudio es el de los **técnicos en Administración de Sistemas Informáticos en Red (ASIR)**, profesionales que desempeñan un papel esencial en las organizaciones, al ser los encargados de mantener, supervisar y optimizar la infraestructura tecnológica. Sus responsabilidades incluyen la configuración y administración de sistemas operativos (Linux y Windows) mediante Bash scripting y PowerShell, la monitorización de las infraestructuras con herramientas como Zabbix para detectar fallos (ej., caídas de servidores), la gestión de bases de datos mediante consultas SQL para generar informes (ej., estadísticas de uso de recursos), y el soporte técnico, que abarca la resolución de incidencias y la gestión de tickets. Estas tareas requieren habilidades técnicas como scripting, gestión de redes, y resolución de problemas, pero también enfrentan desafíos como la alta carga de trabajo operativa y la necesidad de responder rápidamente a incidencias.

En este entorno, herramientas de inteligencia artificial, automatización robótica e integraciones, están comenzando a integrarse en flujos de trabajo reales. Muchas empresas están incorporando automatizaciones para mejorar su eficiencia. Estas herramientas mejoran la eficiencia y permiten a los técnicos enfocarse en tareas estratégicas, como la mejora de la infraestructura o la ciberseguridad. Sin embargo, también presentan riesgos, como la posible reducción del aprendizaje práctico de herramientas tradicionales, la dependencia excesiva de las nuevas herramientas y el desplazamiento de roles operativos si las tareas se automatizan por completo, lo que resalta la necesidad de una adaptación y formación equilibrada.

Tras identificar el problema y el contexto laboral, la siguiente sección analiza en profundidad las necesidades tecnológicas y estratégicas para implementar estas soluciones en un entorno ASIR.

6. Análisis del proyecto

Este apartado analiza en profundidad el impacto y la viabilidad de implementar herramientas de IA y automatización en el entorno laboral del técnico ASIR. Se evalúan sus beneficios, riesgos y requisitos técnicos para entender cómo podrían integrarse en un flujo de trabajo real. Este análisis da soporte a las decisiones que se justificarán en la sección 7 (Diseño del Proyecto) y se basa en los hallazgos obtenidos durante el estado del arte y la investigación metodológica.

6.1. Impacto funcional y beneficios en las áreas de trabajo ASIR

Las herramientas de inteligencia artificial (IA), automatización robótica de procesos (RPA) y plataformas de integración low-code/no-code (LCNC) tienen un impacto significativo en las áreas de trabajo de los técnicos de Administración de Sistemas Informáticos en Red (ASIR). Este análisis evalúa cómo estas herramientas optimizan las tareas diarias en cuatro áreas clave: **administración de sistemas, redes y monitorización, gestión de bases de datos, y soporte técnico**.

Además, se destacan los beneficios operativos que aportan, como la automatización de tareas repetitivas, la reducción de errores humanos y la mejora en la calidad del servicio IT, directamente relacionados con el problema de la sobrecarga operativa identificado en la sección 5.2. Por último, se analiza la relación de estas herramientas con las competencias del ciclo ASIR.

6.1.1. Administración de sistemas (Linux y Windows)

La administración de sistemas operativos, tanto **Linux como Windows**, implica tareas repetitivas que los técnicos ASIR pueden optimizar mediante herramientas de IA como GitHub Copilot y Cursor, descritas en la sección 4.1.1. Estas herramientas impactan positivamente al reducir el tiempo necesario para desarrollar scripts en Bash y PowerShell, utilizados frecuentemente en la configuración de servidores y la gestión de usuarios. Por ejemplo, un técnico puede emplear GitHub Copilot para automatizar la creación de un script que gestione permisos de usuarios en un servidor Linux, lo que permite dedicar más tiempo a tareas estratégicas como la planificación de actualizaciones del sistema.

En entornos Windows, la generación de scripts para tareas administrativas complejas, como la configuración de políticas de grupo en Active Directory (AD), reduce la carga operativa del técnico. El impacto funcional se traduce en una mayor eficiencia, permitiendo a los técnicos ASIR abordar problemas más complejos, como la optimización de recursos del servidor. Un beneficio clave es la mejora en la productividad, que puede incrementarse significativamente, contribuyendo a aliviar la sobrecarga operativa identificada en el diagnóstico GitHub, 2023.

6.1.2. Redes y monitorización

La **monitorización de redes y sistemas** es una tarea esencial para los técnicos ASIR, quienes utilizan herramientas como Zabbix para detectar fallos, como caídas de servidores o problemas de conectividad. Las plataformas LCNC, descritas en la sección 4.1.3, permiten automatizar flujos de trabajo relacionados con la monitorización, mejorando la eficiencia operativa. Por ejemplo, un flujo automatizado puede enviar alertas a Slack cuando se detecta una incidencia crítica, reduciendo el tiempo de respuesta del técnico (n8n, s.f.)

Estas plataformas pueden ahorrar varias horas semanales al automatizar notificaciones, aliviando la carga administrativa de los técnicos ASIR (Zapier, s.f.). Este impacto funcional permite a los técnicos enfocarse en la resolución de problemas críticos, como la configuración de firewalls, en lugar de gestionar manualmente alertas. Un beneficio adicional es la mejora en la calidad del servicio IT, ya que las respuestas automatizadas minimizan el tiempo de inactividad.

6.1.3. Gestión de bases de datos

La **gestión de bases de datos** implica tareas como la generación de informes mediante consultas SQL y la creación de copias de seguridad, que los técnicos ASIR pueden optimizar con soluciones RPA como UiPath y Power Automate, descritas en la sección 4.1.2. UiPath permite automatizar la ejecución de consultas SQL para generar informes de uso de recursos, como estadísticas de espacio en disco, reduciendo el esfuerzo manual (Valtx, 2024). Esto minimiza errores humanos, un problema común en la ejecución manual de consultas complejas (SAP, s.f.).

Power Automate se integra con servicios de Microsoft como Excel, automatizando la exportación de datos a informes compartidos, y puede programarse para realizar backups

automáticos, notificando al técnico en caso de fallo. El impacto funcional es una mayor consistencia en los procesos, ya que las soluciones RPA ejecutan tareas basadas en reglas con alta fiabilidad (Automation Anywhere, s.f.). El beneficio operativo principal es la liberación de tiempo, permitiendo a los técnicos dedicar más esfuerzo a actividades analíticas, como la optimización del almacenamiento de datos, lo que contribuye a reducir la sobrecarga operativa.

6.1.4. Soporte técnico y resolución de incidencias

El **soporte técnico y la resolución de incidencias** generan una alta carga operativa para los técnicos ASIR, especialmente en la gestión de tickets para tareas como el restablecimiento de contraseñas. Herramientas como Zapier y Automation Anywhere (AA), ya descritas en la sección 4, automatizan estos procesos. Zapier puede integrarse con sistemas de gestión de tickets como ServiceNow para enviar notificaciones automáticas a Slack, agilizando la respuesta del técnico (Zapier, s.f.). AA automatiza el restablecimiento de contraseñas en Active Directory, generando informes de los cambios realizados (Automation Anywhere, s.f.).

Este impacto funcional reduce el tiempo dedicado a incidencias básicas, permitiendo a los técnicos enfocarse en problemas más complejos, como la implementación de medidas de ciberseguridad. El beneficio más notable es el ahorro de tiempo, que contribuye directamente a aliviar la sobrecarga operativa, además de mejorar la calidad del servicio IT al garantizar respuestas más rápidas y consistentes a los usuarios finales.

6.1.5. Relación con las competencias del ciclo ASIR

Estas herramientas no reemplazan directamente los conocimientos adquiridos durante el ciclo formativo, sino que los complementan. Por ejemplo, la comprensión de Bash, PowerShell o SQL sigue siendo fundamental para personalizar flujos en herramientas como n8n, o para depurar errores sugeridos por GitHub Copilot. Asimismo, el conocimiento de herramientas como Zabbix es esencial para configurar integraciones con plataformas LCNC, asegurando que las alertas automatizadas sean precisas y relevantes. Las competencias en scripting y gestión de sistemas permiten a los técnicos supervisar y ajustar los procesos automatizados, garantizando su correcta implementación y funcionamiento.

En conjunto, estas tecnologías ofrecen beneficios que permiten transformar el rol del técnico ASIR hacia una posición más estratégica, con mayor enfoque en tareas de

supervisión, optimización de procesos y mejora continua. En el siguiente apartado se abordarán los riesgos, limitaciones y las implicaciones en materia de ciberseguridad que conlleva esta transformación.

6.2. Riesgos, limitaciones y consideraciones de ciberseguridad

La adopción de herramientas de IA, RPA y LCNC en los flujos de trabajo de los técnicos ASIR introduce desafíos que deben analizarse para garantizar una integración equilibrada y segura. Esta subsección examina los riesgos operativos y limitaciones asociados con estas tecnologías, así como los riesgos de ciberseguridad que surgen al automatizar procesos críticos. Se proponen medidas para mitigar estos desafíos, asegurando que las herramientas complementen las competencias de los técnicos ASIR sin comprometer su desarrollo profesional ni la seguridad de los sistemas.

6.2.1. Riesgos y limitaciones operativos

Uno de los riesgos más significativos es el impacto en el **desarrollo de habilidades técnicas** fundamentales de los técnicos ASIR. La automatización de tareas como la escritura de scripts o la gestión de tickets podría reducir las oportunidades de práctica en competencias clave, como el uso manual de Bash, PowerShell o herramientas de monitorización tradicionales. Por ejemplo, si los técnicos dependen constantemente de herramientas de IA para generar scripts, podrían perder fluidez en la resolución manual de problemas, lo que podría ser perjudicial en situaciones donde las herramientas no estén disponibles o fallen. Este riesgo es especialmente relevante para estudiantes de ASIR, quienes están en proceso de adquirir estas habilidades esenciales.

Otra limitación operativa es la **resistencia al cambio** y la curva de aprendizaje asociada con la adopción de estas tecnologías. Aunque las herramientas LCNC están diseñadas para ser accesibles, configurar flujos complejos (como integraciones entre sistemas de monitorización y plataformas de notificación) requiere un nivel de conocimiento técnico que puede ser un obstáculo para técnicos con experiencia limitada. Esta complejidad puede generar frustración y retrasos en la adopción, especialmente en equipos pequeños donde los recursos para formación son limitados.

Además, la dependencia de herramientas externas plantea un riesgo operativo a largo plazo. Muchas de las soluciones analizadas dependen de servicios en la nube, lo que

podría generar problemas si hay interrupciones en la conectividad o si los proveedores cambian sus modelos de precios o funcionalidades. Por ejemplo, las versiones gratuitas o de prueba de estas herramientas suelen tener limitaciones en el número de flujos o integraciones, lo que podría obligar a las organizaciones a incurrir en costes adicionales para versiones premium, un desafío para empresas con presupuestos restringidos.

6.2.2. Riesgos de ciberseguridad y medidas de mitigación

La automatización de procesos críticos en un entorno ASIR introduce riesgos de ciberseguridad que deben ser gestionados cuidadosamente. Uno de los principales riesgos es la **exposición de datos sensibles** durante las integraciones automatizadas. Por ejemplo, un flujo que conecta una herramienta de monitorización con una plataforma de notificación podría transmitir información crítica, como detalles de incidencias o configuraciones del sistema, a través de canales no seguros, aumentando el riesgo de interceptación por parte de actores maliciosos. Este riesgo se agrava si las credenciales de acceso a estas herramientas no están protegidas adecuadamente.

Otro riesgo significativo es la **introducción de vulnerabilidades** a través de procesos automatizados. Por ejemplo, un bot RPA que ejecuta consultas SQL para generar informes podría ser explotado si no se configura con permisos restringidos, permitiendo accesos no autorizados a bases de datos (CyberArk, s.f.). Además, las herramientas de IA que generan scripts podrían introducir errores o vulnerabilidades en el código si los técnicos no supervisan cuidadosamente las sugerencias, un problema que puede ser crítico en configuraciones de servidores o sistemas operativos.

Para **mitigar estos riesgos**, se proponen varias medidas de ciberseguridad. En primer lugar, es fundamental implementar el principio de privilegio mínimo, asegurando que los bots RPA y los flujos LCNC solo tengan acceso a los recursos estrictamente necesarios para su funcionamiento. Por ejemplo, un bot que genera informes SQL debería tener permisos de solo lectura en la base de datos, evitando modificaciones no autorizadas. En segundo lugar, el uso de cifrado para datos en tránsito y en reposo es esencial para proteger la información transmitida durante las integraciones, como las notificaciones automáticas enviadas a través de plataformas de mensajería (CyberArk, s.f.).

Además, se recomienda establecer políticas de autenticación robustas, como el uso de autenticación multifactor (MFA) para acceder a las herramientas basadas en la nube, reduciendo el riesgo de accesos no autorizados. Finalmente, los técnicos ASIR deben

realizar auditorías periódicas de los flujos automatizados para identificar y corregir configuraciones inseguras, como conexiones no cifradas o permisos excesivos, garantizando que las automatizaciones sean seguras y fiables.

6.3. Requisitos técnicos para la integración

La integración efectiva de herramientas de IA, RPA y LCNC en un entorno ASIR requiere una evaluación cuidadosa de los requisitos técnicos necesarios para su implementación. Esta subsección analiza las necesidades de hardware, software y configuración de redes para garantizar que estas herramientas sean compatibles con los sistemas existentes y las competencias de los técnicos ASIR. El análisis se centra en requisitos generales, dejando el diseño técnico específico para la sección 7, y se basa en las características de las herramientas analizadas y las necesidades típicas de un entorno ASIR.

6.3.1. Hardware y software necesarios

Dado que se planea combinar múltiples soluciones, es esencial garantizar que todas puedan convivir e interconectarse adecuadamente. Un primer paso clave es asegurar que existan **interfaces de integración** (APIs o conectores) entre las plataformas seleccionadas (Processmaker, 2022). La mayoría de proveedores ofrecen APIs para conectar sus productos con otros. En nuestro caso, herramientas como Zapier, Power Automate o n8n ya cuentan con centenares de conectores predefinidos para aplicaciones populares, facilitando la comunicación sin necesidad de programación desde cero. Zapier, por ejemplo, soporta la integración de más de 3.000 aplicaciones SaaS, permitiendo flujos entre servicios como Salesforce, Gmail, Basecamp, etc., lo que resulta muy valioso para unir herramientas dispares. Esta amplia compatibilidad significa que parte de los sistemas habituales en entornos ASIR (correo, herramientas IT, CRM, directorio de usuarios, etc.) podrán enlazarse mediante estas plataformas en la arquitectura integrada propuesta.

Cada herramienta considerada impone sus propios requisitos técnicos. En el caso de GitHub Copilot (y de forma similar la IDE con IA Cursor), se necesita integrarlo en los entornos de desarrollo existentes. Ambas soluciones de asistencia de código se integran en el flujo de trabajo de desarrollo existente, por lo que no alteran la arquitectura de TI a nivel de servidores, pero sí requieren considerar aspectos como la política de seguridad del

código (p. ej., evitar subir código sensible a servicios en la nube) y la capacitación de los desarrolladores para aprovecharlas.

En cuanto a las plataformas de automatización RPA como UiPath, estas suelen demandar un **entorno de ejecución dedicado**. UiPath en particular está diseñado para sistemas Windows; requiere instalar componentes como UiPath Studio (diseño de automatizaciones) y UiPath Robot/Assistant (ejecución de automatizaciones) en equipos o servidores. Los requisitos de hardware de un robot de UiPath no son excesivos: se recomienda una CPU de 64 bits de al menos 4 núcleos a 2.4 GHz, 8 GB de RAM y alrededor de 5 GB de espacio en disco (UiPath, 2022). Estas especificaciones garantizan que los bots puedan operar con fluidez. Además, para una integración robusta en un entorno ASIR, habrá que desplegar un Orchestrator (si se opta por la versión Enterprise) o utilizar la versión Cloud Orchestrator de UiPath, lo cual implica disponer de un servidor (o servicio) para coordinar los bots, almacenar credenciales de forma segura y exponer API para lanzar procesos. En términos de software base, los robots UiPath funcionan en Windows 10/11 o Windows Server modernos, y requieren ciertas dependencias (.NET Framework, Visual C++ Redistributable, navegadores soportados para automatización web, etc.) (UiPath, 2022). Los requisitos para Microsoft Power Automate son similares.

Respecto a las herramientas basadas en la nube, como Zapier, Make, n8n y la versión en línea de Power Automate, tienen **requisitos de hardware mínimos** en el lado del cliente, ya que el procesamiento principal se realiza en los servidores del proveedor. Una estación de trabajo estándar para un técnico ASIR, con un procesador Intel i5, 8 GB de RAM y 256 GB de almacenamiento SSD, es suficiente para ejecutar navegadores modernos (como Chrome o Edge) y entornos de desarrollo integrado (IDE) como Visual Studio Code, utilizados para interactuar con estas herramientas. Esta configuración es común en entornos IT y no requiere hardware especializado. Si se requerirán **conocimientos técnicos específicos** en función de que herramienta o software concreto se quiera integrar con la solución LCNC.

6.3.2. Configuración de redes

Es importante considerar requisitos transversales: una **red fiable y segura** para interconectar todos estos componentes. Por ejemplo, si los bots de UiPath se ejecutarán en máquinas virtuales, estas deben tener acceso de red a los sistemas destino (bases de datos, aplicaciones web, etc.) y/o a Internet (para servicios cloud como OpenAI o Zapier).

De igual manera, se necesitan credenciales de servicio para las automatizaciones: un aspecto técnico crítico es habilitar un almacén seguro de contraseñas o tokens (UiPath Orchestrator ofrece Assets para esto, Power Automate usa Connection Credentials, n8n puede utilizar Vaults o variables de entorno cifradas).

El técnico ASIR debe preparar el entorno para **cumplir políticas de seguridad**: por ejemplo, configurar reglas de firewall para permitir comunicación con servicios externos, crear cuentas con permisos acotados para que los bots realicen sus tareas, aislar las máquinas donde corren los robots (quizás en una VLAN específica) y asegurarse de registrar logs (registros) de las acciones automatizadas. También se recomienda que la infraestructura disponga de entornos de pruebas separados (p.ej., un Active Directory de laboratorio, una base de datos de test) donde validar las integraciones antes de pasarlas a producción.

En resumen, los requisitos técnicos abarcan tanto las plataformas software necesarias (IDE compatibles, sistemas operativos soportados, dependencias, cuentas de servicio) como la preparación del entorno (APIs habilitadas, red y hardware suficientes, gestión de credenciales) para que la integración de agentes de IA y automatizaciones sea viable y fluida dentro del ecosistema IT actual.

6.4. Sostenibilidad, escalabilidad y viabilidad de la solución

Al evaluar la solución propuesta, es fundamental analizar si esta puede mantenerse en el tiempo, crecer con las necesidades y realmente implementarse con éxito en un entorno ASIR típico. En términos de **sostenibilidad**, un aspecto clave es el mantenimiento continuo de las automatizaciones y agentes de IA. Las herramientas basadas en la nube, como Zapier y Make o n8n Cloud, son sostenibles mientras el proveedor mantenga el servicio. Si se opta por opciones autoalojadas como n8n, se requiere mantenimiento continuo, lo que puede implicar costos de personal técnico.

Con respecto a las soluciones de RPA empresarial, estas tecnologías no son “instalar y olvidar”: por ejemplo, una vez en producción, los bots RPA requieren mantenimiento constante para asegurar que sigan funcionando conforme cambian los sistemas o interfaces (Medium, 2025). Si una aplicación actualiza su interfaz o un formulario cambia de campo, el flujo automatizado podría fallar y necesitar ajustes. De igual forma, los modelos de IA (como los que usa Github Copilot/Cursor) evolucionan; es posible que

periódicamente haya que actualizar la versión de la herramienta o ajustar la configuración para alinearla con nuevas políticas de la empresa (por ejemplo, filtrado de sugerencias para evitar posibles brechas de código privativo). Esto implica que la solución deberá contar con soporte permanente: el equipo de sistemas (ASIR) debe planificar recursos para monitorizar las automatizaciones, resolver incidencias y aplicar actualizaciones.

Además, es relevante mencionar **sostenibilidad a largo plazo** desde la perspectiva técnica y ambiental. Al automatizar procesos, es posible que a largo plazo se reduzcan errores y retrasos, haciendo la operación más eficiente energéticamente (menos horas de máquinas malgastadas en tareas repetitivas). Sin embargo, también hay que tener en cuenta la huella de nuevas infraestructuras: por ejemplo, ejecutar varios robots 24/7 puede aumentar el consumo eléctrico de servidores; o el uso intensivo de servicios cloud de IA tiene un costo energético en centros de datos remotos. De cualquier modo, dado que el proyecto es teórico y principalmente software, el impacto ambiental directo es bajo, centrándonos más en la sostenibilidad de procesos (que sigan vigentes con el tiempo). Esto implica diseñar automatizaciones flexibles y bien documentadas, para que otro técnico pueda entenderlas y adaptarlas en el futuro si cambian los sistemas.

La **escalabilidad** de la solución es otro factor determinante. Inicialmente, puede que las automatizaciones cubran solo unos pocos procesos repetitivos y los agentes de IA asistan a un puñado de técnicos, pero ¿qué ocurre si se desea ampliar el uso a toda la organización o incrementar significativamente la carga de trabajo automatizada? Con RPA, uno de los desafíos comunes al escalar es que a medida que se añaden más bots, pueden surgir problemas de rendimiento o gestión (Medium, 2025). Nuestra solución debe prever la capacidad de añadir instancias de robots o nodos de ejecución según la demanda (p.ej., utilizar máquinas virtuales escalables o contenedores Docker en el caso de n8n) y balancear la carga de trabajo. Afortunadamente, herramientas empresariales como UiPath ofrecen arquitecturas de alta disponibilidad y escalado horizontal (instalando múltiples robots desatendidos que atienden colas de tareas), y en el caso de servicios cloud (Zapier, Power Automate cloud) la escalabilidad está gestionada por el proveedor – aunque pueden surgir límites de uso o costos crecientes si se exceden ciertos números de ejecuciones mensuales, lo cual toca la viabilidad económica.

Por último, la **viabilidad** de la solución abarca varias dimensiones: técnica, económica y organizativa. Empecemos por la **viabilidad técnica**: ¿Realmente pueden estas herramientas resolver las necesidades identificadas en el entorno ASIR? Del análisis

previo sabemos que hay ciertas limitaciones: por ejemplo, sistemas legacy (con hardware o software antiguo) sin APIs pueden requerir soluciones RPA que son más frágiles, o procesos altamente complejos quizá no sean totalmente automatizables y requieran supervisión humana. Aun así, las tendencias actuales muestran que RPA e IA combinadas permiten abordar un espectro muy amplio de tareas, incluso aquellas que antes quedaban fuera del alcance de la automatización tradicional. Los casos de uso estudiados (como aprovisionamiento de usuarios, gestión de incidencias TI, documentación automática, etc.) son viables de implementar con las herramientas propuestas (Getguru, 2025).

La **viabilidad económica** y de recursos humanos también es crucial. Habría que considerar los costes de licencias: GitHub Copilot es de pago (a nivel empresarial, una suscripción por usuario), UiPath también requiere licencias por robot y quizás por Orchestrator; Zapier tiene planes por número de Zaps y tareas mensuales, etc. Una implementación a gran escala implicaría invertir en estas licencias. No obstante, existen alternativas de código abierto o gratuitas (como n8n en lugar de Zapier, Power Automate Desktop incluido en Windows 10, etc.) que pueden mitigar costos. Un punto a favor es que la mayoría de estas herramientas apuntan a reducción de costes operativos a medio plazo mediante el ahorro de tiempo en tareas rutinarias, por lo que habría que hacer un análisis ROI (Return on Investment): por ejemplo, si automatizar la creación de cuentas ahorra X horas de trabajo de técnicos al mes, ¿compensa el coste de licenciar un robot RPA? Muchos estudios resaltan que RPA fue la tecnología empresarial de más rápido crecimiento en los últimos años, precisamente porque las empresas ven beneficio claro en su adopción (Roboyo, 2021).

La **viabilidad organizativa** se refiere a la aceptación y adaptación dentro del equipo de IT. Un riesgo identificado es la resistencia al cambio por parte de los técnicos, especialmente si perciben estas herramientas como una amenaza a sus puestos (el dilema “¿complemento o competencia?”). Es importante destacar, tal como señalan expertos, que la automatización suele implementarse para apoyar a los empleados y liberarlos de trabajos repetitivos más que para sustituirlos (Medium, 2025). Aun así, la gestión del cambio es esencial: la viabilidad del proyecto dependerá de capacitar a los técnicos ASIR en el uso de estas herramientas, asegurarles que seguirán siendo necesarios (de hecho, más necesarios para supervisar y mejorar las automatizaciones), y fomentar una cultura de innovación. Planes de formación y fases piloto ayudarán a que el equipo gane confianza y vea a estas soluciones como aliadas en su trabajo diario.

En resumen, la solución es viable técnicamente porque aprovecha herramientas maduras y casos de uso concretos del mundo ASIR, escalable siempre que se planifique la infraestructura (multiplicando bots o suscripciones según demanda) y sostenible en la medida que se establezca un buen plan de mantenimiento y gestión del cambio. Los posibles obstáculos (costes, resistencia cultural, integración con legacy) son superables con las estrategias adecuadas. Con todo esto en mente, pasamos a sintetizar las conclusiones del análisis y preparar las bases para el diseño del proyecto.

6.5. Síntesis del análisis y preparación para el diseño

Tras el análisis realizado, podemos concluir que la integración de agentes de IA y automatizaciones RPA/iPaaS en el entorno de un técnico ASIR se perfila más como un complemento potenciador que como una competencia desleal. Cada herramienta evaluada ocupa un nicho específico de tareas: las plataformas RPA (UiPath, Power Automate) y de flujos (Zapier, n8n) brillan en la ejecución automatizada de tareas estructuradas y repetitivas, mientras que los agentes de IA (GitHub Copilot, Cursor) aportan valor en tareas cognitivas o creativas asistidas, como la escritura de código, la generación de documentación o la resolución de problemas mediante sugerencias inteligentes.

El análisis de requisitos mostró que es factible técnicamente integrar todos estos componentes, aunque con las consideraciones de infraestructura y seguridad adecuadas. Asimismo, examinamos la sostenibilidad y vimos que, con un mantenimiento activo y una estrategia de escalado, la solución puede perdurar y crecer dentro de la organización, maximizando beneficios y minimizando riesgos.

Podemos sintetizar las **ventajas** identificadas: la sinergia de RPA + IA promete mayor eficiencia operativa, reducción de errores humanos y liberación de tiempo para que los profesionales de sistemas se enfoquen en tareas de mayor nivel. Estudios y casos reales avalan esta visión, indicando que la combinación de automatización robótica e inteligencia artificial no solo optimiza flujos de trabajo, sino que transforma la forma en que operan las empresas hacia modelos más ágiles y rentables. En el contexto ASIR, esto se traduce en una administración de sistemas más proactiva y menos reactiva.

También resumimos los **desafíos** a afrontar: principalmente, la integración con ciertos sistemas legacy (que puede requerir enfoques híbridos, combinando RPA para unas partes y conectores API para otras), la seguridad (garantizar que ninguna automatización

exponga datos sensibles indebidamente, implementar controles de acceso robustos) y la gestión del cambio organizativo, como se mencionó. Hemos identificado medidas para mitigar estos retos, como delimitar bien responsabilidades y puntos de interacción, facilitando aislar problemas, y adoptar un enfoque incremental, introduciendo primero automatizaciones de procesos muy repetitivos y de bajo riesgo, para luego poder escalar de forma fiable.

De la evaluación de **escalabilidad y sostenibilidad**, derivamos también requisitos de diseño: es necesario que la solución sea modular y extensible. El diseño deberá facilitar el poder agregar nuevos “skills” de IA o nuevas automatizaciones sin rehacer todo el sistema. Igualmente, para sostener el sistema en el tiempo, incorporaremos en el diseño mecanismos de monitorización (por ejemplo, logs centralizados de las tareas automatizadas, alertas si ocurre un fallo) y de actualización (una forma sencilla de reemplazar un script generado por IA con una versión revisada por un humano, por ejemplo).

En resumen, el análisis nos proporciona un mapa claro de qué necesitamos: una infraestructura capaz de alojar bots RPA (en Windows, orquestados adecuadamente), conectores con servicios cloud (mediante Zapier o n8n) para integrar aplicaciones web, asistentes de IA integrados en las herramientas de los técnicos (IDE con GitHub Copilot/Cursor) y todo ello gobernado bajo políticas de seguridad y buenas prácticas de TI. Con esta información, pasamos a la fase de diseño del proyecto, donde definiremos la arquitectura conceptual de la solución, detallando cómo se conectarán estos componentes en un flujo de trabajo coherente y cómo se garantizarán las propiedades deseadas (escalabilidad, seguridad, eficacia).

7. Diseño del proyecto

Tras el análisis exhaustivo realizado en la sección anterior, donde se evaluó el impacto funcional, los riesgos, los requisitos técnicos y la viabilidad de integrar agentes de IA y herramientas de automatización en el entorno ASIR, esta sección se centra en el diseño de una solución conceptual. El objetivo principal de este diseño es abordar directamente la sobrecarga operativa identificada en el diagnóstico (sección 5.2), proponiendo una arquitectura tecnológica que actúe como un complemento eficaz para los técnicos ASIR.

El diseño considera criterios clave: compatibilidad con herramientas tradicionales del temario ASIR (Zabbix, PowerShell, SQL), facilidad de adopción para técnicos con conocimientos básicos, capacidad de supervisión técnica para garantizar un control humano, y reducción efectiva de la carga operativa mediante la automatización de procesos repetitivos. Dado el carácter de investigación del proyecto, el diseño se enfocará en modelar cómo interactuarían los componentes en un entorno ASIR típico, más que en esquemas físicos detallados. Este diseño se basa en el análisis realizado en la sección 6 (Análisis del proyecto), centrándose en este caso en la estructura técnica y operativa, preparando el terreno para un despliegue teórico que se propondrá en la sección 8 (Despliegue y pruebas).

7.1. Arquitectura general de la solución

La solución propuesta se organiza en tres capas funcionales que interactúan entre sí y con las herramientas tradicionales del entorno ASIR, formando un sistema integrado que automatiza flujos de trabajo y asiste a los técnicos en tareas técnicas. Cada capa tiene un propósito específico y se conecta mediante APIs, conectores predefinidos y scripts personalizados, garantizando una comunicación fluida y eficiente.

1. Capa de **asistencia inteligente** (GitHub Copilot, Cursor). Esta capa se integra directamente en los entornos de desarrollo utilizados por los técnicos ASIR (por ejemplo, Visual Studio Code). Su propósito es acelerar el desarrollo y depuración de scripts (Bash, PowerShell), consultas (SQL), ficheros de configuración (YAML, JSON) y otras tareas basadas en código. Genera sugerencias de código contextuales, completa fragmentos, traduce comentarios en lenguaje natural a código funcional y ayuda a identificar errores o proponer optimizaciones. Su interacción es directa con el técnico a través del IDE.
2. Capa de **automatización de procesos y orquestación** (RPA - UiPath, AA, PA). Esta capa se encarga de automatizar tareas estructuradas y repetitivas que involucren la interacción con interfaces gráficas de usuario (GUIs) de aplicaciones que no exponen APIs fácilmente (sistemas legacy, aplicaciones de escritorio), o para ejecutar secuencias complejas de pasos basados en reglas definidas. Ideal para

tareas como la entrada de datos masiva, la reconciliación de información entre sistemas no conectados o la ejecución de procesos en entornos Windows. Puede ser disparada por eventos (monitorizados por la Capa 3), por horarios programados, o manualmente por un técnico.

Interactúa directamente con aplicaciones de escritorio, navegadores web, y puede ejecutar scripts locales (PowerShell, VBScript). Se comunica con la Capa 3 a través de APIs (ej., API de UiPath Orchestrator, Conectores de Power Automate) para recibir instrucciones o devolver resultados.

3. **Capa de integración** (Make, Zapier, n8n). Esta última capa actúa como el sistema nervioso central de la automatización. Conecta diferentes aplicaciones y servicios a través de sus APIs o webhooks (mensaje automatizado que se envía a una aplicación externa cuando ocurre un evento), orquestando flujos de trabajo que cruzan múltiples sistemas. Es ideal para la automatización basada en eventos (ej., una alerta de Zabbix, un nuevo ticket, un email recibido). Gestiona la lógica del flujo (condicionales, bucles, transformaciones de datos) y la comunicación entre servicios.

Se conecta a las APIs de herramientas ASIR (ej., Zabbix API para obtener datos de alertas), servicios cloud (Microsoft 365, Google Workspace, Slack, Teams), bases de datos (SQL), y sistemas de ticketing (Jira, ServiceNow). Puede disparar o ser disparada por la Capa 2 (RPA) a través de APIs/conectores y recibir/enviar datos hacia/desde ella. También puede interactuar con scripts personalizados.

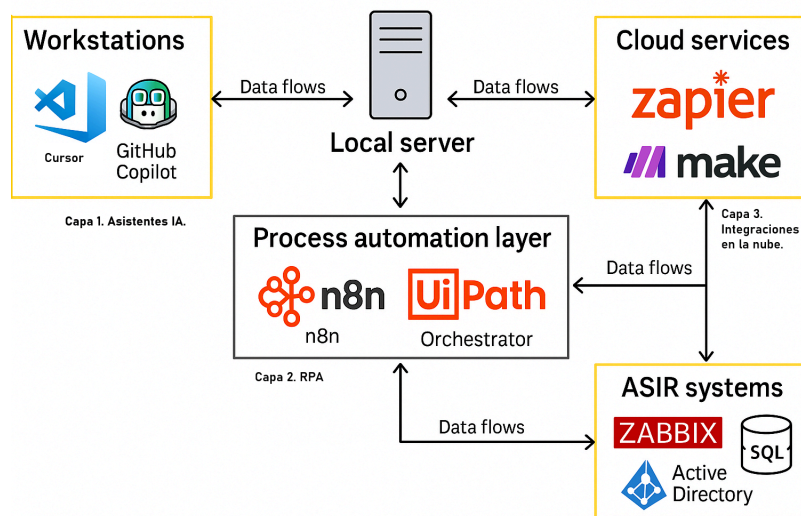
Algunas de estas herramientas son versátiles y podrían actuar en distintas capas, por ejemplo n8n se podría utilizar en la capa 2 como orquestador (figura 3) o Power Automate en la 3, sirviendo como herramienta de integración en la nube.

Las tres capas **se integran con las herramientas estudiadas en el temario ASIR**. Por ejemplo, Zabbix envía alertas a n8n mediante su API, PowerShell es ejecutado por UiPath para tareas administrativas en Windows, y consultas SQL son gestionadas por Power Automate para generar informes. Un diagrama conceptual (figura 3) incluiría: estaciones de trabajo (con Visual Studio Code y GitHub Copilot), un servidor local (con n8n y UiPath Orchestrator), servicios en la nube (Zapier, Make), y sistemas ASIR (Zabbix, AD,

bases de datos SQL), conectados mediante flujos de datos bidireccionales. La potencia de la arquitectura reside en la interacción fluida entre capas.

Figura 3.

Diagrama con las diferentes capas de AI y automatizaciones.



7.2. Flujos de trabajo automatizados propuestos

A continuación, se describen tres flujos de trabajo automatizados que abordan tareas clave de los técnicos ASIR, diseñados para ser implementados con las herramientas seleccionadas. Estos flujos son factibles según la **documentación oficial** de las herramientas y se enfocan en la interacción técnica, sin repetir los beneficios operativos ya mencionados en la sección 6.1.

1. Monitorización de redes y servidores.

Este flujo automatiza la respuesta a incidencias detectadas por Zabbix, una herramienta de monitorización común en ASIR (figura 4).

1. 1 Una alerta de Zabbix (herramienta tradicional) es capturada por n8n (Capa 3) vía webhook. N8n analiza la alerta, la enriquece con datos de una base de datos

SQL (tradicional) y crea un ticket vía API. Simultáneamente, notifica al equipo ASIR en Slack (servicio cloud).

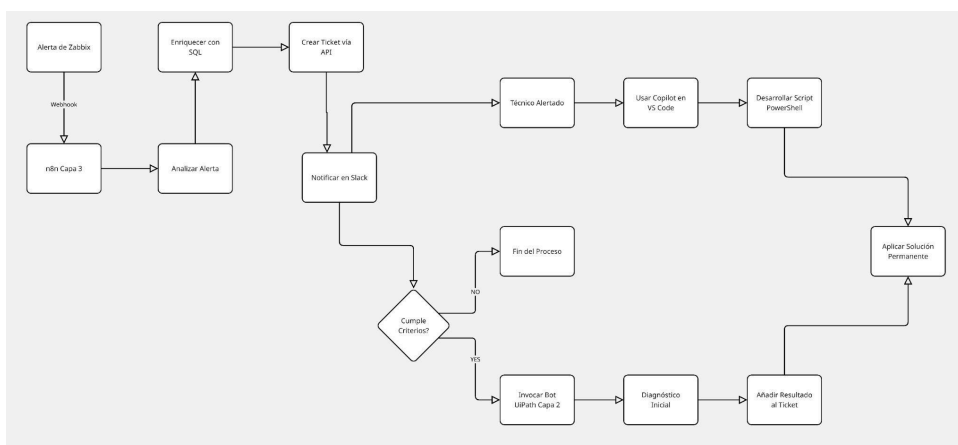
1.2 Si la alerta cumple ciertos criterios, n8n invoca a un bot de UiPath (Capa 2) a través de la API de Orchestrator para realizar un diagnóstico inicial en el servidor afectado (ej., ejecutar comandos PowerShell). El resultado se añade al ticket.

1.3 Mientras tanto, un técnico, alertado por Slack, utiliza GitHub Copilot (Capa 1) en su VS Code para desarrollar un script PowerShell (tradicional) que aplique una solución permanente, basándose en la información recopilada en el ticket.

Este flujo es viable porque cada herramienta utilizada (Zabbix, n8n, UiPath, GitHub Copilot, Slack) tiene capacidades documentadas que soportan las acciones descritas.

Figura 4.

Diagrama con automatización para monitorización de redes y servidores.



2. Gestión de bases de datos.

Este flujo automatiza la generación diaria de informes de uso de recursos a partir de una base de datos SQL (figura 5).

1.1 Power Automate Desktop, instalado en una estación de trabajo Windows, se programa para ejecutarse cada mañana a las 8:00. El flujo comienza ejecutando una consulta SQL (por ejemplo, "SELECT * FROM disk_usage WHERE date = CURRENT_DATE") en una base de datos MySQL, utilizando un conector ODBC (Open Database Connectivity) configurado con permisos de solo lectura.

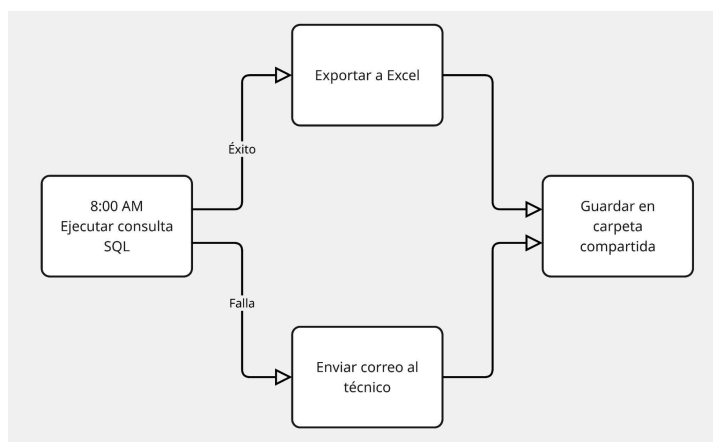
1.2 Los resultados se exportan a un archivo Excel mediante la acción nativa de Power Automate para manejar hojas de cálculo. Si la consulta falla (por ejemplo, por un error de conexión), Power Automate envía un correo al técnico vía Outlook, notificando el problema.

1.3 El archivo Excel se guarda en una carpeta compartida para revisión.

Este flujo es factible porque Power Automate Desktop soporta conectores SQL y acciones de Excel, y su integración con Outlook es nativa.

Figura 5.

Diagrama con automatización para gestión de base de datos.



3. Soporte técnico.

Este flujo automatiza la gestión de tickets de restablecimiento de contraseñas, una tarea común en soporte técnico (figura 6).

1.1 Zapier conecta un sistema de tickets como ServiceNow con Active Directory: cuando se crea un ticket de tipo "restablecimiento de contraseña", Zapier detecta el evento mediante su integración con ServiceNow y activa un bot de UiPath.

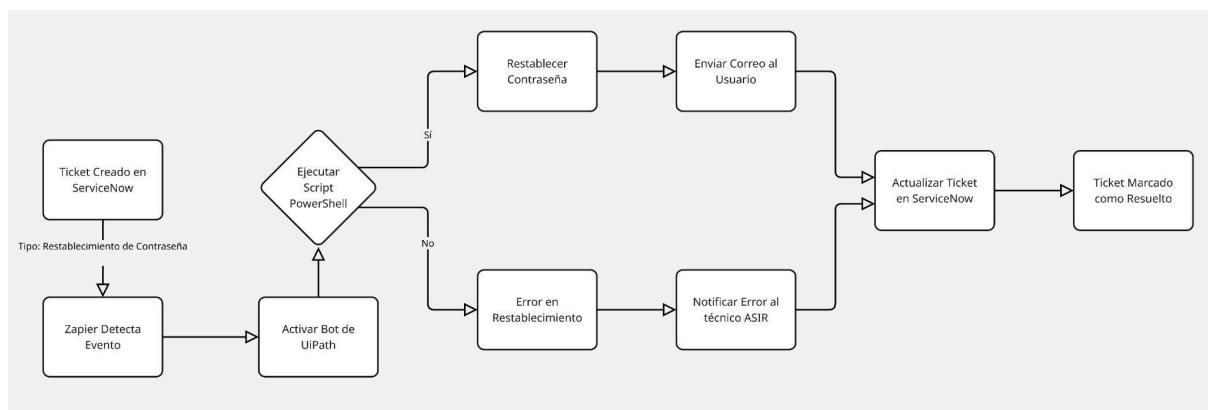
1.2 El bot, gestionado por UiPath Orchestrator, ejecuta un script PowerShell (por ejemplo, "Set-ADAccountPassword") para restablecer la contraseña, utilizando credenciales seguras almacenadas en Orchestrator. Si hubiera un error, se notificaría al técnico ASIR.

1.3. Una vez completado, Zapier envía un correo al usuario final desde Gmail, notificándole el cambio, y actualiza el ticket en ServiceNow como "resuelto".

Este flujo es viable porque Zapier tiene conectores para ServiceNow y Gmail, y UiPath puede ejecutar scripts PowerShell para gestionar Active Directory.

Figura 6.

Diagrama con automatización para soporte técnico.



7.3. Seguridad, gobierno y mantenimiento del sistema

Esta subsección sirve como complemento a la 6.4. Detalla las configuraciones de seguridad, la gestión de cambios y el mantenimiento del sistema, asegurando que la solución sea robusta y sostenible a nivel técnico.

Para empezar, es importante definir la **seguridad y políticas de acceso**. Se aplica el principio de privilegio mínimo: cada robot o flujo solo accede a los sistemas que necesita. Además, se usan mecanismos como, OAuth2 o claves API rotativas para accesos externos, MFA y cifrado en herramientas SaaS (Zapier, Make) y Vaults (cajas fuertes) o almacenes cifrados (n8n, UiPath Orchestrator) para gestionar credenciales de forma segura. Los firewalls se configuran para permitir solo tráfico autorizado (por ejemplo, puerto 443 para Zapier), bloqueando accesos externos no deseados. Por último, se habilita HTTPS mediante un certificado SSL, configurando el servidor local correctamente y asegurando que todas las comunicaciones estén cifradas.

Con respecto a la **gestión de cambios, versiones y actualizaciones**, se propone un repositorio Git (usando GitHub) donde se almacenen los scripts generados por GitHub Copilot y los flujos de n8n exportados como archivos JSON. Esto permite versionar los flujos y scripts, facilitando la reversión en caso de errores. Las actualizaciones de herramientas se planifican trimestralmente: UiPath Orchestrator y Power Automate Desktop se actualizan a las versiones más recientes, verificando primero en un entorno de pruebas para evitar interrupciones. n8n, alojado localmente, se actualiza asegurando que los flujos existentes sean compatibles con la nueva versión. Los modelos de IA se actualizan automáticamente por el proveedor, pero los técnicos deben revisar las sugerencias de código para garantizar que cumplen con las políticas de seguridad de la organización, evitando la inclusión de código sensible.

Finalmente, se detalla la **modularidad** para poder incluir **nuevos flujos**. El diseño por capas es modular para permitir la incorporación de nuevos flujos o herramientas. Añadir un nuevo flujo en n8n para integrar otra herramienta (ej., un nuevo sistema de monitorización) no requiere modificar los flujos existentes si no interactúan directamente. UiPath Orchestrator permite escalar el número de bots, asignando nuevas tareas mediante colas de trabajo. Esta modularidad asegura que el sistema pueda evolucionar sin rehacer la arquitectura, enfocándose en la flexibilidad técnica.

7.4. Plan de formación para los técnicos ASIR

Para garantizar la adopción de la solución, se propone un **plan de capacitación específica**, para los técnicos ASIR, dividido en tres fases, impartida cada una en una semana, para que sea progresivo:

1. Semana 1. **Sensibilización y visión general**:
 - a. Presentación de la arquitectura diseñada y los objetivos (alineamiento con la reducción de carga operativa).
 - b. Demostración de los flujos de trabajo automatizados clave y los beneficios esperados.
 - c. Discusión abierta sobre el impacto en el rol ASIR (enfaticando el "complemento" y la oportunidad de enfocarse en tareas más complejas).
 - d. Presentación del plan de formación completo.

2. Semana 2. **Capacitación específica por capas:**

- a. Módulo Asistentes IA: Uso efectivo de GitHub Copilot/Cursor en VS Code. Técnicas de prompting, revisión crítica del código generado, implicaciones de seguridad. (Prácticas con scripts Bash/PowerShell/SQL).
- b. Módulo LCNC (n8n/Make/PA Cloud): Introducción a la plataforma elegida. Cómo entender/monitorizar flujos existentes. Creación de flujos sencillos (ej., notificación Slack desde email). Conexión a APIs básicas. Manejo de errores simple. (Ejercicios prácticos en entorno de pruebas).
- c. Módulo RPA (UiPath/PA Desktop): Comprensión de qué es RPA y cuándo usarlo. Cómo monitorizar la ejecución de bots desde Orchestrator/Power Platform. Cómo interactuar con bots atendidos (si aplica). (Enfoque en el uso y supervisión, no necesariamente desarrollo inicial).

3. Semana 3. **Seguridad y buenas prácticas:**

- a. Políticas de seguridad específicas para el uso de las nuevas herramientas.
- b. Manejo seguro de credenciales en flujos y bots.
- c. Proceso de gestión de cambios y reporte de incidencias en las automatizaciones.
- d. Cuándo y cómo escalar problemas o solicitar nuevas automatizaciones.

Además de estas 3 fases/semanas iniciales, se realizará una **cuarta fase**, que incluirá un **soporte continuo**. Se establecerá de un canal de soporte interno (ej., canal de Teams/Slack) para dudas, donde también se incluirá documentación de forma accesible (guías rápidas, FAQs, grabaciones de formación). Por último, se llevarán a cabo sesiones periódicas (mensuales/trimestrales) para compartir experiencias, nuevos casos de uso, y actualizaciones de las plataformas.

Este plan asegura que los técnicos ASIR adquieran las habilidades necesarias para usar y supervisar la solución, promoviendo una adopción efectiva sin requerir conocimientos avanzados previos. El objetivo es que no solo sean usuarios pasivos, sino actores clave en la mejora y mantenimiento de estas soluciones.

Como resumen, el diseño se ha realizado con la seguridad por diseño (security by design), integrando las protecciones en cada capa en vez de añadirlas después. Un técnico ASIR encontrará en este diseño una estructura familiar donde se prioriza el principio de mínimo privilegio, la segregación de funciones, la monitorización constante y la capacidad de respuesta ante incidentes. Esto garantiza que la incorporación de IA y automatización no comprometa la robustez de la infraestructura, sino que, al contrario, la refuerce al eliminar en lo posible el error humano y aplicar procedimientos consistentes. Con la arquitectura y estos detalles definidos, ya podemos detallar cómo podríamos desplegar y probar esta solución de manera teórica en la siguiente sección, asegurando que cada componente funcione según lo previsto antes de una hipotética implantación real.

8. Despliegue y pruebas

Esta sección describe la estrategia de despliegue gradual de la solución diseñada y el plan de pruebas exhaustivas para validar su correcto funcionamiento. Dado que el proyecto es teórico y no se implementará físicamente, se plantea un plan simulado de cómo se llevaría a cabo la puesta en marcha en un entorno real y cómo se verificarían todos los componentes. El enfoque es garantizar que, de materializarse el proyecto, se seguirían las mejores prácticas de despliegue en entornos corporativos ASIR, con mínima interrupción de servicios y máxima fiabilidad en los resultados.

8.1. Plan de despliegue teórico

El despliegue de la solución seguiría un esquema escalonado en fases, mitigando riesgos y permitiendo adaptación sobre la marcha. Las fases propuestas son:

1. **Fase 0: Preparación del entorno:** antes de “encender” cualquier automatización, se acondiciona la infraestructura. En esta etapa se instalan y configuran las herramientas: por ejemplo, montar el servidor de n8n en la red de pruebas, instalar UiPath Studio y Orchestrator en servidores designados (o configurar Power Automate Desktop en las máquinas necesarias), preparar las cuentas de servicio en AD, y obtener las suscripciones requeridas (activar GitHub Copilot para los usuarios del equipo, contratar plan de Zapier adecuado).

Asimismo, se implementa el almacén de credenciales seguro y se cargan en él los secretos iniciales. Se ejecutan pequeñas pruebas de conectividad: comprobar que n8n puede llamar a un API externo, que Orchestrator UiPath se comunica con su robot, que GitHub Copilot se integra en los IDEs. Esta fase es fundamentalmente técnica y no interactúa aún con procesos de negocio.

2. **Fase 1: Despliegue en entorno de pruebas** (pilotaje): con todo listo, se pasa a desplegar la solución en un entorno de pruebas que emule la producción. Aquí se importan los flujos diseñados (p. ej., importar los flows de Power Automate a un entorno de prueba, desplegar los workflows de n8n en su instancia de test, publicar los procesos UiPath en Orchestrator con conexión a sistemas de prueba). Se inicia un periodo de prueba piloto en áreas acotadas: por ejemplo, elegir 1 o 2 procesos (casos de uso) para automatizar inicialmente y verlos funcionar end-to-end en este entorno simulado.

Los miembros del equipo de TI participarían simulando ser usuarios finales para generar triggers de prueba (llenar formularios ficticios, crear incidencias de prueba, etc.). El objetivo es evaluar el impacto y resultado en pequeño antes de una adopción más amplia (Convotis, s.f.). En esta fase se monitorea muy de cerca el comportamiento: se verifica que los bots realizan las acciones correctas, que las integraciones entregan los datos esperados, y se miden tiempos de ejecución. También se busca detectar cualquier anomalía o efecto no previsto. Por ejemplo, podría descubrirse que un flujo se ejecuta demasiado lento o que GitHub Copilot sugiere código ineficiente.

Estos hallazgos permiten refinar las automatizaciones. Durante el pilotaje, se reúnen comentarios de los técnicos: ¿les resulta útil Github Copilot en su día a día? ¿La notificación automatizada en Teams del bot es comprensible? Etc. Con base en esto, se ajustan detalles en la configuración o se proporciona capacitación adicional si se ve confusión.

3. **Fase 2: Ajustes y validación iterativa:** es muy probable que tras las primeras pruebas surjan pequeñas correcciones. Esta fase es iterativa: se aplican cambios en los flujos/robots (por ejemplo, agregar un retraso si un sistema tarda en responder,

mejorar una condición lógica) y se vuelve a probar. Se realizan pruebas de estrés en el entorno de test: lanzar múltiples procesos simultáneamente para ver cómo aguanta la orquestación, introducir datos extremos o casos límite. También se prueban fallos inducidos: desconectar temporalmente un servicio para ver si el sistema reacciona según lo diseñado (¿Lo reintenta? ¿Avisa a un humano?).

Por otro lado, se hace una evaluación de seguridad en pruebas: un analista de seguridad podría intentar, por ejemplo, usar un token expirado en el webhook para asegurarse de que es rechazado, o simular un usuario malicioso ingresando datos malformados en un formulario para ver si la automatización los filtra adecuadamente.

Todos estos tests aseguran que la solución cumple los requisitos no funcionales (seguridad, robustez). Si algo falla, se regresa al diseño/implementación para corregir y se vuelve a validar. Esta fase continúa hasta que el equipo esté convencido de que la solución en pruebas se comporta correctamente en un entorno controlado.

4. **Fase 3: Despliegue en producción controlado:** una vez validados los pilotos, se planifica el despliegue en el entorno real de producción. Esto no se hace de golpe para todos los procesos; adoptamos una estrategia de implementación progresiva (rollout). Por ejemplo, primero se habilita el flujo de alta de empleados automatizado, pero manteniendo el proceso manual paralelo como respaldo.

Durante un periodo, cada vez que se use la automatización, el resultado será revisado manualmente por un técnico para confirmar que todo salió bien (doble ejecución, aunque es redundante, da confianza). Conforme la automatización demuestra fiabilidad, se dejará de lado el proceso manual. Luego se pasará a automatizar otro proceso (p. ej., la generación de informes semanales), y así sucesivamente.

Este enfoque por lotes minimiza el riesgo: si algo va mal, solo afectará al proceso recién implementado y se puede revertir rápidamente a modo manual. Durante este despliegue inicial en producción, se sigue un plan de comunicación claro: los usuarios finales o implicados (por ejemplo, el departamento de RRHH que recibe informes) deben saber que ahora ciertas tareas las realiza un “agente automático” y cómo interactuar (p. ej., si ven un usuario creado por “System” no alarmarse porque

es el bot). Igualmente, el equipo de TI tendrá una lista de comprobación diaria para monitorear las primeras ejecuciones en real, verificando logs y resultados.

5. **Fase 4: Expansión y transferencia:** una vez que los módulos iniciales funcionan bien en producción, se procede a ampliar la solución a otros procesos de la empresa, siguiendo la misma mecánica de pruebas en laboratorio, piloto, producción paulatina.

Además, se contempla la transferencia de conocimiento y cierre del proyecto: documentar detalladamente la arquitectura final desplegada, las configuraciones de cada herramienta, y formar al personal de soporte que se encargará del mantenimiento a largo plazo (Roboyo, 2021). El diseño teórico sugiere incluso la creación de un sub departamento especializado, dentro del departamento de TI, tal como recomiendan especialistas RPA para asegurar la continuidad y mejora constante de estas iniciativas. Este sub departamento sería responsable de revisar periódicamente los flujos, proponer nuevas automatizaciones y velar por el cumplimiento de estándares. Si bien esto excede el despliegue inmediato, forma parte de la “puesta en marcha” prolongada en el tiempo para que el proyecto sea vivo y adaptable.

En todo el despliegue, se tiene preparado un plan de contingencia: si algún componente falla catastróficamente o genera resultados erróneos, se aislará ese componente. Por ejemplo, si el bot RPA de cierto proceso empieza a cometer errores graves, se desactiva su ejecución (el orquestador lo omite) y los técnicos retomarán esa tarea manualmente hasta solucionar el problema. De esta manera, el negocio no se detiene por confiar demasiado en la automatización. El éxito del despliegue se medirá en que, al final del proyecto, los procesos automatizados funcionen con un índice de error muy bajo (idealmente 0 en casos manejados, y con planes de recuperación para errores inesperados) y que los usuarios/técnicos estén satisfechos con el nuevo flujo de trabajo.

8.2. Plan de pruebas y validación

El plan de pruebas cubre distintos niveles para asegurar que cada pieza y el conjunto integrado cumplen con los requisitos. A continuación, se detallan los tipos de pruebas a realizar y cómo se aplicarían a nuestra solución teórica:

1. **Pruebas unitarias de componentes:** Aunque en RPA/automatización las “unidades” son flujos de trabajo, aplicamos el concepto de probar cada elemento de forma aislada. Por ejemplo, probar un script generado por GitHub Copilot por separado para verificar que realiza la tarea esperada (si es un script PowerShell para crear usuarios, probarlo en un AD de pruebas con diferentes inputs). Igualmente, probar un workflow de n8n de forma aislada con datos simulados, o un proceso UiPath ejecutándolo en su estudio paso a paso. Estas pruebas unitarias buscan detectar errores lógicos o de sintaxis en cada bloque antes de integrarlo con otros.

Además, se realizarán pruebas unitarias de las reglas de negocio: por ejemplo, alimentar el flujo de Power Automate con un caso donde debería bifurcar por una condición y ver si toma la rama correcta. Para las unidades que involucran IA (GitHub Copilot), se hace revisión manual: por ejemplo, inspeccionar que el código sugerido no contenga llamadas a endpoints (puntos de acceso en aplicaciones o APIs) incorrectos o posibles vulnerabilidades. Esta etapa se da mayormente durante el desarrollo, antes del pilotaje.

2. **Pruebas de integración:** Aquí el foco es verificar que las interacciones entre componentes funcionan correctamente. Un caso de prueba típico sería: “Cuando se crea un registro en el sistema X, el flujo Y debe tomarlo, pasar por Zapier, invocar a UiPath y finalmente actualizar el sistema Z”. Simulamos el evento inicial (creación del registro) y luego rastreamos todo el pipeline (serie de procesos o pasos). Comprobamos que: Zapier recibió el webhook y lanzó la acción, n8n procesó los datos y llamó al bot, el bot ejecutó su tarea y devolvió resultado, etc. Nos fijamos en puntos de integración específicos, por ejemplo: ¿los datos mapeados entre Zapier y n8n mantienen el formato correcto (encoding (codificación), decimales, etc.)? ¿El token de autenticación que envía Power Automate al llamar la API de Orchestrator (orquestador) es válido y se acepta?

Estas pruebas pueden requerir instrumentar un poco el código para loggear mensajes de depuración en cada frontera. Dado que nuestro proyecto tiene varias integraciones, dedicaremos suficiente tiempo a esta fase. Si se detectan fallos, se itera: por ejemplo, si un campo obligatorio no estaba siendo enviado de Zapier a n8n, se corrige el mapeo y se repite la prueba.

Las pruebas de integración también incluyen verificar comportamientos concurrentes: lanzar dos procesos diferentes a la vez que comparten un recurso (p.ej., dos bots intentando escribir en la misma base de datos) y confirmar que no se molestan – aquí entran en juego bloqueos controlados o colas si necesario.

3. **Pruebas de rendimiento y carga:** Aunque la solución no es una aplicación transaccional de alto volumen, es importante probar su comportamiento bajo carga elevada o prolongada. Simularemos escenarios donde, por ejemplo, se generen 100 solicitudes de una tacada (quizá usando scripts para crear eventos masivos) y observar si el orquestador las encola apropiadamente o si alguno de los servicios se ralentiza. También mediremos los tiempos de ejecución de ciertas tareas para establecer límites base: p.ej., crear un usuario con todo el proceso toma 30 segundos; este será el baseline (línea de partida) y se comprueba que no aumente significativamente con carga moderada. Otra prueba de estrés sería mantener bots corriendo durante horas ejecutando operaciones repetitivas para ver si hay fugas de memoria o acumulación de procesos.

Dado que algunas partes dependen de servicios cloud (GitHub Copilot, Zapier), aquí también se podría observar limitaciones externas: por ejemplo, Zapier free plan solo permite X tareas por minuto. Si en pruebas de carga se alcanza un límite, se consideraría subir de plan o rediseñar para no golpear tanto el servicio.

El desempeño global es importante para la escalabilidad: estas pruebas nos dirán hasta cuántas operaciones diarias podemos hacer con la configuración actual y sirven para planificar aumento de recursos en el futuro.

4. **Pruebas de seguridad:** Ya en integración mencionamos algunas, pero se podría hacer un pentesting (prueba de penetración para identificar vulnerabilidades) específico. Se probará que los controles de acceso funcionan: intentar que un

usuario sin permisos active un flujo crítico (debe ser rechazado), o que un flujo no permita elevar privilegios (por ejemplo, que un bot que solo debe leer no pueda accidentalmente escribir si modifica la entrada).

Si hay interfaces web expuestas (como un webhook), probar inyecciones de SQL/Script en los datos enviados para asegurar que no comprometen nada (lo ideal es que esos datos se traten solo como entrada y no generen acciones peligrosas sin validación). También se comprobará la persistencia de datos sensibles: revisar logs para ver que no se están guardando contraseñas o información personal en texto plano. Un aspecto a probar es el fail-safe (sistema a prueba de fallos): desconectar deliberadamente la red de un bot a mitad de operación y ver que hace el sistema (no debería, por ejemplo, quedar media transacción en un estado inseguro; el Orchestrator debería marcar error y en la siguiente corrida retomar desde inicio).

Todas estas pruebas se documentan con sus resultados, y cualquier hallazgo de vulnerabilidad se corrige antes de mover a producción.

5. **Pruebas de aceptación de usuario (UAT):** Si bien los “usuarios” de esta solución son mayormente el propio equipo IT, los técnicos ASIR y quizás algunos departamentos beneficiarios, es importante realizar una validación con los stakeholders finales. En esta etapa, se invitaría a personal de, digamos, recursos humanos o soporte a que interactúen con el sistema en modo prueba: por ejemplo, que RRHH inicie un proceso de alta de empleado usando la nueva forma (que desencadena la automatización) y verifique que el resultado final (usuario dado de alta, email enviado, etc.) cumple sus expectativas. O que el equipo de soporte vea llegar notificaciones de incidentes creados automáticamente y evalúe si la información es adecuada.

Estos test de usuario garantizan que la solución no solo funciona internamente, sino que encaja con las operaciones reales. Dado el carácter teórico, no obtendremos tal feedback, pero en un despliegue real se recogerían comentarios y se harían ajustes menores (por ejemplo, cambiar el formato de un correo de notificación si los usuarios piden más claridad).

6. **Prueba de todo el sistema (End-to-End):** Finalmente, se realiza una o varias pruebas completas de escenarios de negocio de principio a fin, sin intervención manual intermedia. Esto equivale a un ensayo general: se escoge un día y se ejecuta, por ejemplo, todo el proceso de onboarding ficticio desde que RRHH introduce datos hasta que el usuario ficticio inicia sesión con su nueva cuenta, involucrando cada paso automatizado. O resolver un incidente TI de prueba donde participa la automatización. Se supervisa en tiempo real y se verifican todos los outputs. Si todo funciona correctamente, se declara que la solución ha pasado la fase de pruebas.

8.3. Criterios de éxito y aceptación

En esta subsección se detallan brevemente los ciertos criterios para considerar que la fase de despliegue y pruebas ha sido exitosa. Para considerar que las pruebas han sido superadas, definimos algunos criterios medibles:

1. **Funcionalidad:** Cada caso de uso automatizado produce el resultado esperado sin errores en al menos 9 de cada 10 ejecuciones (90% éxito mínimo, ideal 100% en casos normales).
2. **Rendimiento:** Los tiempos de ejecución se mantienen dentro de límites aceptables (por ejemplo, que ningún proceso automatizado tarde más que haciéndolo manualmente, salvo actividades asíncronas que antes no eran posibles).
3. **Seguridad:** No se han encontrado vulnerabilidades explotables y todas las acciones quedan registradas adecuadamente para auditoría.
4. **Usabilidad:** Los usuarios y técnicos involucrados comprenden el funcionamiento y están de acuerdo en que mejora su flujo de trabajo (esto es subjetivo, pero puede medirse vía encuestas de satisfacción post-UAT). Si alguno de estos criterios no se cumple, se documenta como deficiencia y se corrige antes de avanzar.

En caso de que las pruebas revelen que algún objetivo original no se logró del todo, se podría reevaluar el alcance: por ejemplo, si cierto proceso resultó demasiado complejo de automatizar con el nivel de fiabilidad deseado, quizás se decida mantenerlo manual por ahora (lo cual es un hallazgo válido en un proyecto de investigación: saber qué no conviene automatizar aún). Estas decisiones formarían parte de las conclusiones del proyecto.

Con este riguroso plan de despliegue y pruebas teórico, se pretende demostrar que la integración de agentes IA y automatizaciones propuesta sería técnicamente sólida y beneficiosa. Habríamos identificado y corregido problemas, afinado el rendimiento, y validado que la solución cumple con los requisitos planteados en el análisis. Todo ello sin haber puesto en riesgo la operativa real en ningún momento. Con esto, el proyecto de investigación queda prácticamente cerrado, quedando ya solamente lugar para las conclusiones y vías futuras.

9. Conclusiones

Esta sección resume los resultados clave del proyecto, evalúa el grado de consecución de los objetivos iniciales, reflexiona sobre el proceso de desarrollo y los aprendizajes obtenidos, reconoce las limitaciones inherentes al trabajo realizado y concluye sobre la contribución general del proyecto.

El **objetivo principal** del presente proyecto fue evaluar el impacto de los agentes de inteligencia artificial, las herramientas de automatización de procesos y las plataformas de integración en los roles profesionales del sector IT, con un enfoque particular en los técnicos de Administración de Sistemas Informáticos en Red (ASIR). La pregunta central que ha guiado este estudio ha sido determinar si estas tecnologías emergentes actúan principalmente como un complemento que potencia las capacidades de los técnicos ASIR, o como una competencia directa que amenaza su estabilidad laboral.

Tras un análisis exhaustivo del estado del arte, la evaluación de herramientas específicas, la consideración de riesgos y beneficios, y el diseño de una arquitectura conceptual integrada, este estudio concluye que, cuando se implementan de manera estratégica y reflexiva, estas tecnologías funcionan **predominantemente como un complemento potenciador** para los técnicos ASIR cualificados. Lejos de reemplazar de inmediato al profesional, actúan como asistentes que automatizan tareas repetitivas, agilizan flujos de trabajo y reducen errores humanos, permitiendo al técnico enfocarse en labores de mayor nivel. Del estudio teórico se desprende también que estas herramientas son viables y funcionalmente impactantes en el entorno ASIR actual.

La viabilidad quedó reflejada en la disponibilidad creciente de estas soluciones (muchas integradas en entornos de desarrollo o como servicios en la nube) y en casos de

uso alineados con tareas reales de administración. Por supuesto, se identificaron **riesgos a gestionar**: depender demasiado de sugerencias de IA sin verificación podría introducir configuraciones erróneas o inseguras; asimismo, una automatización mal diseñada podría propagar un fallo a gran velocidad. Por ello, el técnico sigue siendo esencial para validar, ajustar y mantener estas automatizaciones.

Ahora bien, considerar estas tecnologías únicamente como complemento sin evaluar su faceta de “competencia” sería incompleto. En ciertos ámbitos, las automatizaciones pueden sustituir tareas concretas tradicionalmente realizadas por humanos. Un robot de RPA bien configurado puede ejecutar 24/7 procesos repetitivos sin descanso, y un asistente de código avanzado puede generar configuraciones estándares en segundos, lo que podría reducir la necesidad de intervención humana en esas áreas. Esto implica que, para tareas altamente estandarizadas y rutinarias, las organizaciones podrían depender más de herramientas que de personal junior. No obstante, esta posible competencia se traduce más bien en una **transformación del rol**: el técnico ASIR deja de ser un “ejecutor manual” de procedimientos para convertirse en un orquestador y supervisor de sistemas automatizados. En el proyecto se analizaron los riesgos y viabilidad de dicha transformación. Se concluyó que, si bien existe el riesgo de desplazamiento de ciertas funciones, prevalece la oportunidad de que los técnicos evolucionen asumiendo funciones de mayor valor. De este modo, la competencia se mitiga cuando el profesional adopta la herramienta como aliada.

Esto nos lleva a la conclusión de que **las necesidades de formación están cambiando**: será necesario capacitar a los futuros administradores de sistemas no solo en los fundamentos tradicionales (sistemas operativos, redes, seguridad), sino también en el uso eficaz de asistentes de IA y plataformas de automatización, así como en las áreas que no puede cubrir la automatización (diseño de arquitecturas, toma de decisiones estratégicas, trato con usuarios, etc.).

Los cinco **objetivos específicos** también se cumplieron prácticamente en su totalidad, y son los propios de un proyecto de investigación. Las limitaciones en la consecución de tales objetivos, también fueron debidas a la naturaleza teórica del proyecto. Por ejemplo, aunque se analizó el impacto en las competencias de los técnicos ASIR (sección 6), no se pudo evaluar empíricamente su efecto en la eficiencia o calidad de los procesos mediante pruebas prácticas. La ausencia de un entorno real impidió medir métricas concretas, como el tiempo ahorrado en la resolución de incidencias. Asimismo, el objetivo de determinar implicaciones laborales a largo plazo quedó parcialmente cubierto, ya

que se identificaron habilidades clave (como la gestión de flujos LCNC), pero no se pudo realizar un estudio práctico que lo apoyase. Estas limitaciones se compensaron con un enfoque teórico robusto, utilizando fuentes actuales y un plan de despliegue y pruebas simulado.

La realización de este proyecto ha sido una experiencia enriquecedora a nivel académico y personal. Como estudiante de ASIR, investigar un tema tan actual como la IA y la automatización me ha permitido comprender mejor las tendencias que definirán mi futuro profesional. Me ha sorprendido el potencial de estas herramientas para transformar tareas operativas, pero también he tomado conciencia de la importancia de mantener un equilibrio entre automatización y habilidades técnicas tradicionales. El proceso de investigación, aunque arduo, me ha enseñado a gestionar fuentes diversas y a estructurar un análisis complejo, habilidades que serán útiles en mi carrera. Además, diseñar una solución teórica me ha motivado a considerar cómo podría implementarla en un entorno real, reforzando mi interés por la automatización como un campo de especialización.

10. Vías futuras

Si bien este proyecto ofrece una primera aproximación teórica al impacto de los agentes de IA y automatizaciones en el rol del técnico ASIR, presenta limitaciones importantes que abren oportunidades para trabajos futuros. En primer lugar, se trató de un análisis teórico sin simulación práctica: por falta de tiempo y recursos no se pudo montar un “laboratorio” donde probar efectivamente estas herramientas en escenarios reales de administración de sistemas. En segundo lugar, el alcance se centró en agentes asistentes de IA (asistentes de programación, plataformas RPA/LCNC), no abarcando los “agentes operadores” autónomos capaces de tomar acciones sin intervención humana. Estas limitaciones sugieren varias líneas de mejora e investigación a desarrollar:

- **Pruebas en entornos reales o de laboratorio:** Implementar experimentos controlados donde un técnico ASIR realice ciertas tareas con y sin ayuda de estas herramientas, comparando métricas como tiempo invertido, tasa de errores, rendimiento y resultados. Por ejemplo, se podría simular el alta masiva de usuarios en un directorio activo: un grupo de administradores lo haría manualmente o con scripts tradicionales, mientras otro usaría Power Automate o UiPath. Analizando los

resultados se obtendrían datos concretos sobre la eficacia y eficiencia que aporta la automatización en situaciones prácticas, validando (o matizando) las conclusiones teóricas del proyecto.

- **Exploración de agentes de IA autónomos:** Ampliar el estudio hacia los llamados agentes inteligentes autónomos u operadores automáticos emergentes. Herramientas de nueva generación como AutoGPT, AgentGPT o Manus, representan IAs que pueden encadenar tareas por sí solas (por ejemplo, diagnosticar un problema en un servidor web investigando logs y aplicando parches de forma autónoma). Investigaciones futuras podrían evaluar cómo estos agentes autónomos encajarían en el entorno ASIR, qué tipos de tareas podrían delegarse en ellos y cuáles serían los riesgos y controles necesarios. Esto implicaría considerar aspectos como la supervisión humana de las decisiones de la IA, mecanismos de seguridad para evitar acciones dañinas y criterios para confiar tareas críticas a una inteligencia no humana.
- **Evaluación comparativa con más herramientas:** Extender el conjunto de herramientas analizadas para comparar enfoques y fabricantes diversos. Por un lado, incluir otros asistentes de código basados en IA (p. ej., Amazon CodeWhisperer, Tabnine u opciones de código abierto) para ver si ofrecen ventajas o resultados distintos a GitHub Copilot/Cursor en el ámbito de scripting para sistemas. Por otro lado, considerar más soluciones de RPA y automatización de distintos proveedores (como Blue Prism, u orquestadores de código abierto adicionales) así como plataformas LCNC emergentes. Este estudio comparativo más amplio aportaría una visión más generalizable, identificando qué conclusiones son independientes de la herramienta específica y qué consideraciones particulares deben tenerse según la elección de plataforma.
- **Estudios con profesionales ASIR en activo:** Complementar la perspectiva académica con la opinión y experiencia de técnicos reales. Se propone realizar encuestas, entrevistas o grupos de enfoque con administradores de sistemas en distintas organizaciones (desde PYME hasta gran empresa) para recopilar

información sobre el grado de adopción actual de estas herramientas. Preguntas como “¿Utiliza asistentes de IA como Copilot en sus tareas diarias?”, “¿Ha implantado flujos de trabajo con herramientas tipo Zapier/Make en su departamento?”, “¿Percibe estas tecnologías como una amenaza para su puesto o como una ayuda?”, proporcionarían datos valiosos.

El análisis de estas respuestas permitiría contrastar las expectativas teóricas vs. la realidad del sector, identificando también barreras de adopción (falta de conocimiento, resistencias culturales, limitaciones presupuestarias) que no se abordan en un trabajo teórico. Asimismo, la retroalimentación de profesionales podría guiar recomendaciones más precisas sobre formación y políticas de implantación de IA en unidades de sistemas.

- **Incorporación de simulaciones y formación especializada:** Como línea adicional, se sugiere investigar cómo integrar estos contenidos en la formación ASIR y medir su impacto. Por ejemplo, diseñar módulos prácticos dentro del currículo donde alumnos utilicen Copilot para elaborar scripts de administración o configuren procesos en Power Automate/n8n para resolver casos prácticos. Posteriormente, evaluar si esos alumnos desarrollan mejor ciertas competencias (velocidad de desarrollo, capacidad de automatización) que aquellos formados sin estas herramientas. Esta vía permitiría validar la viabilidad educativa de incluir IA y automatización en la enseñanza, asegurando que los técnicos egresados estén mejor preparados para el entorno laboral venidero. También podría explorarse el desarrollo de bancos de pruebas o simuladores que emulen incidentes de sistemas resolubles tanto manualmente como con ayuda de IA, para afinar en un entorno seguro las estrategias de colaboración humano-IA antes de su aplicación en producción.

En resumen, las vías futuras pasan por llevar este estudio del plano teórico al práctico, profundizando el alcance hacia nuevas tecnologías autónomas y contrastando con la realidad profesional. Realizar estas investigaciones complementarias reforzará el conocimiento sobre cómo convivir con agentes de IA en entornos de administración de sistemas, proporcionando bases más sólidas para maximizar su papel como complemento y minimizar cualquier efecto adverso en el empleo y la calidad del servicio IT.

11. Bibliografía/Webgrafía

Automation Anywhere. (s.f.). *¿Qué es la automatización robótica de procesos (RPA)?*

<https://www.automationanywhere.com/la/rpa/robotic-process-automation>

CyberArk. (s.f.). *¿Qué es la automatización robótica de procesos (RPA)?*

<https://www.cyberark.com/es/what-is/robotic-process-automation/>

Convotis. (s.f.). *Combinación de RPA e IA para mejorar la productividad.*

<https://www.convotis.com/es/5-pasos-para-aprovechar-todo-el-potencial-de-los-datos-de-tu-empresa/>

Daily.dev. (2024). *Cursor AI: The AI-powered code editor changing the game.*

<https://daily.dev/blog/cursor-ai-everything-you-should-know-about-the-new-ai-code-editor-in-one-place>

Gartner. (2024). *Intelligent Agents in AI Really Can Work Alone. Here's How.*

<https://www.gartner.com/en/articles/intelligent-agent-in-ai>

Gartner. (2024). *Top 10 strategic technology trends for 2025.*

<https://www.gartner.com/en/articles/top-technology-trends-2025>

Getguru. (2025). *Cómo usar Zapier: una guía completa.*

<https://www.getguru.com/es/reference/how-to-use-zapier-a-comprehensive-guide>

GitHub. (2023). *The economic impact of GitHub Copilot on developer productivity.*

<https://github.blog/news-insights/research/the-economic-impact-of-the-ai-powered-developer-lifecycle-and-lessons-from-github-copilot/>

IMF. (2024). *La economía mundial transformada por la inteligencia artificial ha de beneficiar a la humanidad.*

<https://meetings.imf.org/es/IMF/Home/Blogs/Articles/2024/01/14/ai-will-transform-the-global-economy-lets-make-sure-it-benefits-humanity>

McKinsey & Company. (2023). *The state of AI in 2023: Generative AI's breakout year.*

<https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2023-generative-ais-breakout-year>

Medium. (2025). What businesses need to know before automating at scale.

<https://vikranttomar.medium.com/the-hidden-challenges-of-scaling-rpa-what-businesses-need-to-know-before-automating-at-scale-54ebd97e1416>

n8n. (s.f.). *ITOps Workflow Automation Software & Tools - n8n.* <https://n8n.io/itops/>

Leadpicker. (s.f.). *Learn more about Make (Integromat), providing top-notch services in the field of Make (Integromat).*

<https://www.leadpicker.com/ai-tools/make-integromat>

Processmaker. (2022). *Hacer inteligente la automatización de los negocios: Un marco para combinar DPA, RPA e IA.*

<https://www.processmaker.com/es/blog/intelligent-business-automation-framework-combining-dpa-rpa-and-ai/>

Roboyo. (2021). *Solving the Unsolved: How to scale RPA Successfully.*

<https://roboyo.global/blog/rpa-scalability-challenges-what-they-are-and-how-we-can-solve-them/>

SAP. (s.f.). *¿Qué es la automatización robótica de procesos (RPA)?*

<https://www.sap.com/latinamerica/products/technology-platform/process-automation/what-is-rpa.html>

UiPath. (2022). *Robot - hardware and software requirements.*

<https://docs.uipath.com/robot/standalone/2022.10/user-guide/hardware-and-software-requirements>

Valtx. (2024). *Principales software RPA que debes conocer para automatizar los procesos en tu empresa.*

<https://www.valtx.pe/blog/top-5-principales-software-rpa-que-debes-conocer-para-automatizar-los-procesos-en-tu-empresa>

Zapier. (s.f.). *IT automation solutions from Zapier.* <https://zapier.com/solutions/it>

12. Glosario

Glosario de términos ordenado alfabéticamente.

- AA (Automation Anywhere): Plataforma de software empresarial para la Automatización Robótica de Procesos (RPA).
- Active Directory (AD): Servicio de directorio desarrollado por Microsoft para redes de dominio Windows. Se utiliza comúnmente para la gestión de usuarios, equipos y políticas en una red empresarial.
- AD: Ver Active Directory.
- Agente IA / Agente Inteligente: Sistema de software que puede percibir su entorno, procesar información y actuar de forma autónoma o semiautónoma para alcanzar objetivos específicos. En el contexto del proyecto, incluye asistentes (como Copilot) y potencialmente operadores autónomos.
- API (Application Programming Interface - Interfaz de Programación de Aplicaciones): Conjunto de definiciones y protocolos que permite que diferentes aplicaciones de software se comuniquen entre sí.
- ASIR (Administración de Sistemas Informáticos en Red): Ciclo Formativo de Grado Superior en España enfocado en la administración y mantenimiento de sistemas informáticos y redes. También se refiere al profesional técnico con esta formación.
- Automatización Robótica de Procesos (RPA): Tecnología que utiliza "robots" de software para automatizar tareas repetitivas basadas en reglas, interactuando con sistemas digitales de la misma manera que lo haría un humano.
- Backup: Copia de seguridad de datos realizada para poder restaurarlos en caso de pérdida.
- Baseline (Línea base): Punto de referencia o estado conocido con el que se comparan mediciones futuras, especialmente en pruebas de rendimiento para evaluar cambios.
- Bash scripting: Creación de scripts (archivos de comandos) utilizando el lenguaje de intérprete de comandos Bash, común en sistemas operativos tipo Unix (como Linux).
- Blue Prism: Otra plataforma de software empresarial para RPA, mencionada como ejemplo no analizado en detalle.
- Bot (Robot de software): Aplicación de software que realiza tareas automatizadas. En el contexto de RPA, imita acciones humanas en interfaces digitales.

- CFGS (Ciclo Formativo de Grado Superior): Nivel de estudios de formación profesional en España.
- Commit (Git): Acción en el sistema de control de versiones Git que guarda los cambios realizados en el código o archivos dentro de un repositorio.
- Copilot: Ver GitHub Copilot.
- CRM (Customer Relationship Management - Gestión de Relaciones con Clientes): Estrategias y tecnologías utilizadas por las empresas para gestionar y analizar las interacciones con sus clientes.
- Cursor: Entorno de Desarrollo Integrado (IDE) potenciado por IA, diseñado para facilitar la escritura y edición de código.
- Encoding (Codificación): Proceso de convertir datos de un formato a otro, relevante para asegurar la compatibilidad y correcta interpretación de caracteres entre sistemas.
- Endpoint: Punto final de comunicación en una red. En el contexto de APIs, es la URL específica donde se puede acceder a un recurso o servicio.
- Fail-safe: Sistema o mecanismo diseñado para prevenir daños o fallos graves en caso de que ocurra un error.
- Firewall (Cortafuegos): Sistema de seguridad de red que monitoriza y controla el tráfico de red entrante y saliente basándose en reglas de seguridad predeterminadas.
- Flujo de trabajo (Workflow): Secuencia de pasos o tareas automatizadas o manuales necesarias para completar un proceso específico.
- Framework: Marco de trabajo; estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que sirve de base para la organización y desarrollo de software.
- Gamma App: Herramienta online para crear presentaciones interactivas.
- Gantt (Diagrama de Gantt): Herramienta visual de gestión de proyectos que ilustra el cronograma del proyecto, mostrando las tareas y su duración en el tiempo.
- Git: Sistema de control de versiones distribuido, ampliamente utilizado para el seguimiento de cambios en el código fuente durante el desarrollo de software.
- GitHub: Plataforma de desarrollo colaborativo basada en Git, que ofrece alojamiento de repositorios, control de versiones y otras herramientas para desarrolladores.

- GitHub Copilot: Asistente de programación basado en IA desarrollado por GitHub y OpenAI, que sugiere código y funciones completas en tiempo real dentro del editor de código.
- Google Drive: Servicio de almacenamiento de archivos en la nube ofrecido por Google.
- Google Scholar: Buscador de Google enfocado en literatura académica y científica.
- GUI (Graphical User Interface - Interfaz Gráfica de Usuario): Tipo de interfaz de usuario que permite a los usuarios interactuar con dispositivos electrónicos a través de iconos gráficos y indicadores visuales.
- Hardware: Componentes físicos de un sistema informático (ej., CPU, RAM, disco duro).
- HTTPS (Hypertext Transfer Protocol Secure): Versión segura del protocolo HTTP, que utiliza cifrado (normalmente SSL/TLS) para proteger la comunicación entre el navegador web y el servidor.
- IA (Inteligencia Artificial): Campo de la informática dedicado a la creación de sistemas capaces de realizar tareas que normalmente requieren inteligencia humana, como el aprendizaje, la resolución de problemas y la toma de decisiones.
- IA Agéntica: Subcampo de la IA centrado en la creación de agentes autónomos (sistemas) que pueden planificar y ejecutar secuencias de acciones para lograr objetivos complejos con mínima intervención humana.
- IDE (Integrated Development Environment - Entorno de Desarrollo Integrado): Aplicación de software que proporciona servicios integrales a los programadores para el desarrollo de software (ej., editor de código, depurador, compilador).
- IMF (International Monetary Fund - Fondo Monetario Internacional): Organismo internacional mencionado en la introducción en relación con el impacto de la IA en el empleo.
- iPaaS (Integration Platform as a Service - Plataforma de Integración como Servicio): Soluciones basadas en la nube que permiten desarrollar, ejecutar y gobernar flujos de integración entre diferentes aplicaciones y servicios.
- IT/TI (Information Technology - Tecnologías de la Información): Sector relacionado con el uso de computadoras, software, redes y otros equipos físicos e infraestructura para gestionar y procesar información.

- ITOps (IT Operations - Operaciones de TI): Conjunto de procesos y servicios administrados por el departamento de TI de una organización para gestionar su infraestructura tecnológica.
- JSON (JavaScript Object Notation): Formato ligero de intercambio de datos, fácil de leer y escribir para los humanos, y fácil de interpretar y generar para las máquinas.
- Jira: Herramienta popular para la gestión de proyectos, seguimiento de errores y gestión de incidencias.
- LCNC (Low-Code/No-Code): Enfoque de desarrollo de software que permite crear aplicaciones y automatizaciones con mínima o ninguna codificación manual, a menudo utilizando interfaces visuales y componentes preconstruidos.
- Legacy (Sistema heredado): Sistema informático, tecnología o aplicación antigua que sigue en uso a pesar de existir tecnologías más nuevas y eficientes.
- Logs (Registros): Archivos que registran eventos, acciones o mensajes generados por un sistema informático o una aplicación, útiles para monitorización, diagnóstico y auditoría.
- Make (anteriormente Integromat): Plataforma de integración y automatización visual (iPaaS/LCNC) que permite conectar aplicaciones y automatizar flujos de trabajo.
- MFA (Multi-Factor Authentication - Autenticación Multifactor): Método de seguridad que requiere que el usuario proporcione dos o más factores de verificación para acceder a un recurso.
- Microsoft 365: Suite de servicios de suscripción de Microsoft que incluye aplicaciones de Office, servicios en la nube como OneDrive, Teams y Exchange Online.
- n8n: Plataforma de automatización de flujos de trabajo, disponible como servicio en la nube o autoalojable (open source), a menudo considerada más técnica que Zapier o Make.
- OBS Studio: Software libre y de código abierto para grabación de vídeo y transmisión en vivo.
- ODBC (Open Database Connectivity): Estándar de API para acceder a sistemas de gestión de bases de datos (DBMS).
- Onboarding: Proceso de incorporación de nuevos empleados a una organización.
- Open source (Código abierto): Software cuyo código fuente está disponible públicamente para que cualquiera pueda verlo, modificarlo y distribuirlo.

- Orchestrator (Orquestador): Componente central en algunas plataformas RPA (como UiPath Orchestrator) que gestiona, programa, monitoriza y despliega los robots de software.
- OAuth2: Protocolo de autorización estándar abierto que permite a las aplicaciones obtener acceso limitado a cuentas de usuario en servicios HTTP.
- PA (Power Automate): Servicio de Microsoft para crear flujos de trabajo automatizados entre aplicaciones y servicios. Incluye versiones en la nube y de escritorio (Power Automate Desktop).
- Pentesting (Prueba de penetración): Ataque simulado autorizado a un sistema informático para evaluar su seguridad e identificar vulnerabilidades.
- Pipeline (Tubería/Cauce): En el contexto de procesos y datos, se refiere a una serie de etapas conectadas por donde fluye el trabajo o la información, siendo la salida de una etapa la entrada de la siguiente.
- Power Automate: Ver PA.
- PowerShell: Framework de automatización de tareas y lenguaje de scripting desarrollado por Microsoft, basado en .NET. Ampliamente utilizado para la administración de sistemas Windows.
- Prompting (Ingeniería de prompts): Proceso de diseñar y refinar las instrucciones (prompts) dadas a un modelo de IA para obtener la respuesta deseada.
- PYME (Pequeña y Mediana Empresa): Clasificación de empresas basada en su tamaño (número de empleados, volumen de negocio).
- ROI (Return on Investment - Retorno de la Inversión): Métrica financiera utilizada para evaluar la rentabilidad de una inversión.
- Rollout (Despliegue progresivo): Estrategia de implementación de un nuevo sistema o funcionalidad de manera gradual a grupos de usuarios o partes del entorno, en lugar de hacerlo todo de golpe.
- RPA: Ver Automatización Robótica de Procesos.
- SaaS (Software as a Service - Software como Servicio): Modelo de distribución de software donde las aplicaciones son alojadas por un proveedor y puestas a disposición de los usuarios a través de internet mediante suscripción.
- Scripting: Proceso de escribir scripts, que son programas o secuencias de comandos diseñados para automatizar tareas o controlar el comportamiento de otro software.

- Security by design (Seguridad por diseño): Enfoque de desarrollo de software y sistemas que integra consideraciones de seguridad desde las primeras etapas del diseño, en lugar de añadirlas posteriormente.
- ServiceNow: Plataforma basada en la nube que ofrece software de gestión de servicios de TI (ITSM), operaciones de TI (ITOM) y gestión de negocios de TI (ITBM).
- Slack: Plataforma de comunicación empresarial que ofrece mensajería instantánea, canales temáticos y otras herramientas de colaboración.
- SQL (Structured Query Language - Lenguaje de Consulta Estructurada): Lenguaje estándar utilizado para gestionar y consultar bases de datos relacionales.
- SSL (Secure Sockets Layer): Protocolo criptográfico estándar (aunque ahora mayormente reemplazado por TLS) que proporciona comunicaciones seguras a través de una red. A menudo se usa junto con HTTPS.
- Stakeholder (Parte interesada): Individuo, grupo u organización que puede afectar o ser afectado por las acciones o políticas de una empresa o proyecto.
- Teams (Microsoft Teams): Plataforma de comunicación y colaboración unificada de Microsoft, que combina chat, videoconferencias, almacenamiento de archivos e integración de aplicaciones.
- UAT (User Acceptance Testing - Pruebas de Aceptación de Usuario): Fase final del proceso de pruebas de software donde los usuarios finales prueban el sistema en un entorno "real" o simulado para verificar si cumple con sus necesidades y expectativas antes de su aceptación.
- UiPath: Una de las principales plataformas de software empresarial para la Automatización Robótica de Procesos (RPA).
- TLS (Transport Layer Security): Protocolo criptográfico que asegura comunicaciones en redes, sucesor de SSL. Usado en HTTPS, emails, y APIs.
- Vault (Caja fuerte / Almacén seguro): Componente o sistema diseñado para almacenar y gestionar de forma segura información sensible como contraseñas, claves API o certificados.
- VBScript (Visual Basic Scripting Edition): Lenguaje de scripting desarrollado por Microsoft, a menudo utilizado en entornos Windows para automatización de tareas.
- Visual Studio Code (VS Code): Editor de código fuente ligero pero potente, desarrollado por Microsoft, que soporta múltiples lenguajes de programación y extensiones.

- VLAN (Virtual Local Area Network - Red de Área Local Virtual): Técnica para segmentar lógicamente una red física en múltiples redes virtuales independientes.
- VS Code: Ver Visual Studio Code.
- Webhook: Mecanismo automatizado que permite que una aplicación notifique a otra aplicación en tiempo real cuando ocurre un evento específico, enviando datos a una URL determinada.
- YAML (YAML Ain't Markup Language): Lenguaje de serialización de datos legible por humanos, utilizado comúnmente para archivos de configuración.
- Zabbix: Herramienta de monitorización de código abierto para redes, servidores, máquinas virtuales y servicios en la nube.
- Zapier: Plataforma online (iPaaS/LCNC) que permite conectar diferentes aplicaciones web y automatizar tareas entre ellas sin necesidad de código.