UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



"DISEÑO DE ARQUITECTURA PARA AUTOMATIZACIÓN DE CARGA DE FACTURAS CON MICROSERVICIOS EN LA NUBE"
CASO: EVOL SERVICES

SEBASTIAN IGNACIO BENAVIDES CASTILLO

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía (Mentor): Luis Hevia
Profesor Correferente: Edison Delgado – Evol Services

Octubre - 2026

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Diseño de arquitectura para automatización de carga de facturas con microservicios en la nube.

Nombre del candidato(a): Sebastián Ignacio Benavides Castillo.

Carrera / Grado: Ingeniería Civil Informática.

Campus: Casa Central, Valparaíso. **Departamento:** Informática.

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Luis Hevia, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción):**

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

_____

_____

_____

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 27-10-2026. **Firma:** _____

Estudiante o Candidato(a):

Fecha: 27-10-2026 **Firma:** _____

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Para mi familia, amigos y todos los que estuvieron en el proceso, aquellos que aportaron directamente y quienes, solo con el hecho de estar, dieron el ánimo necesario.

AGRADECIMIENTOS

A mis padres, Yerty Castillo y Erwin Benavides, por el apoyo y el ánimo que me dieron para dar el paso de dejar mi ciudad natal e ir a Valparaíso en busca de un futuro mejor, y por los sacrificios que hicieron para que eso fuera posible. Gracias por su paciencia y comprensión durante todos los años de la carrera, incluidos los extras.

En el desarrollo de esta memoria, agradezco al profesor Luis Hevia, mi profesor guía, quien, quizás sin saberlo, me dio la pauta necesaria para lograr avances reales y esforzarme en ellos. A Edison Delgado, Coordinador de Desarrollo y Tecnología y mi jefe, por creer en mí desde la práctica profesional y darme la oportunidad de quedarme y seguir aprendiendo. A Martín León, Coordinador de GCP, por compartir su conocimiento del negocio, por su ayuda con los requerimientos y por darme ánimo para terminar.

Por último, al área de Desarrollo y Tecnología y a todo el equipo humano de Evol Services, porque es en las personas donde se forma el buen ambiente de trabajo.

RESUMEN

Resumen— Evol Services valida los cobros de sus clientes libres con los datos de unas 800 facturas mensuales, en 30 formatos. Su carga en EMER era el único paso manual. Una automatización por reglas redujo el esfuerzo de 60 a 28-33 horas-persona al mes, pero falla en cerca del 30% de las facturas y su mantención consume más de un tercio de ese esfuerzo.

Esta memoria diseña, implementa y valida una arquitectura de microservicios en la nube que automatiza la extracción y carga de esos datos: cinco servicios *serverless* en AWS coordinados por colas, extracción con un modelo de lenguaje y una validación determinística que deriva a revisión toda factura dudosa.

Con 100 facturas reales en el ambiente de calidad, la solución alcanzó una precisión de 98%, frente a 91% del *bot*; incorporar un formato tomó de 2 a 4 horas, frente a cerca de 25, y el esfuerzo mensual estimado fue de 10 a 16 horas-persona, frente a 28-33 del *bot* y 60 del manual. Cumple las metas de los ocho escenarios de calidad, lo que la hace viable para reducir el esfuerzo y los errores y reemplazar progresivamente al *bot*.

Palabras Clave—Arquitectura de software; Microservicios; Computación sin servidor; Modelos de lenguaje; Extracción de datos.

ABSTRACT

Abstract— *Evol Services validates the payments of its free-market electricity clients using data from about 800 monthly invoices in 30 formats. Loading them into EMER was the only manual step. A rule-based automation cut the effort from 60 to 28-33 person-hours per month, but it fails on about 30% of the invoices, and its maintenance takes more than a third of that effort.*

This thesis designs, implements, and validates a cloud microservices architecture that automates the extraction and loading of that data: five serverless services on AWS coordinated by queues, extraction with a language model, and a deterministic validation that routes any doubtful invoice to manual review.

With 100 real invoices in the quality assurance environment, the solution reached an accuracy of 98%, compared with 91% for the bot; adding a new format took 2 to 4 hours, compared with about 25, and the estimated monthly effort was 10 to 16 person-hours, compared with 28-33 for the bot and 60 for the manual procedure. It met the goals of all eight quality scenarios, making it viable for reducing effort and errors and progressively replacing the bot.

Keywords— *Software architecture; Microservices; Serverless computing; Large language models; Data extraction.*

GLOSARIO

ADD: *Attribute-Driven Design*. Método de diseño de arquitecturas que deriva las decisiones a partir de los atributos de calidad requeridos.

ADR: *Architecture Decision Record*. Registro breve y fechado de una decisión arquitectónica, su contexto, las alternativas evaluadas y sus consecuencias.

API: *Application Programming Interface*. Interfaz que permite a un programa solicitar servicios o datos a otro.

AWS: Amazon Web Services. Proveedor de servicios de computación en la nube utilizado por Evol Services.

BPMN: *Business Process Model and Notation*. Notación estándar para modelar procesos de negocio.

C4: Modelo de documentación de arquitecturas de *software* en cuatro niveles de abstracción: contexto, contenedores, componentes y código.

DLQ: *Dead-letter queue*. Cola que recoge los mensajes que agotaron sus reintentos, para su inspección y reproceso posterior.

DX: Empresa distribuidora de energía eléctrica. Transporta la energía hasta el punto de suministro y emite su propia factura.

EC2: *Elastic Compute Cloud*. Servicio de AWS que provee servidores virtuales.

EMER: Sistema interno de Evol Services donde se registran los datos de facturas, mediciones y contratos, y donde se elaboran los reportes.

FaaS: *Functions as a Service*. Modelo en que la unidad de despliegue es una función que el proveedor ejecuta en respuesta a un evento, sin que el cliente administre servidores.

FIFO: *First in, first out*. Cola que entrega los mensajes en el mismo orden en que llegaron, si uno falla, los siguientes de su grupo esperan a que se resuelva. La solución la usa para cargar las facturas en EMER de a una.

GCP: Equipo de Gestión y Control de Procesos de Evol Services, responsable de los insumos de los reportes y de su elaboración.

h-p: Horas-persona. Unidad de esfuerzo equivalente al trabajo de una persona durante una hora.

IaaS: *Infrastructure as a Service*. Modelo en que el proveedor entrega máquinas virtuales, almacenamiento y red, y el cliente administra el sistema operativo y lo que corre sobre él.

IAM: *Identity and Access Management*. Servicio de AWS que define qué puede hacer cada componente sobre cada recurso.

IDP: *Intelligent Document Processing*. Categoría de soluciones que encadenan reconocimiento del contenido, interpretación y validación para convertir documentos en datos consumibles por otros sistemas.

ISO/IEC 25010: Norma que define un modelo de calidad del producto de *software*. La edición de 2023 establece nueve características, cada una con subcaracterísticas.

JSON: *JavaScript Object Notation*. Formato de texto para el intercambio de datos estructurados.

LGSE: Ley General de Servicios Eléctricos (DFL N° 4/20.018, de 2006), que regula el sector eléctrico chileno.

LLM: *Large Language Model*. Modelo de lenguaje de gran escala, entrenado para procesar e interpretar texto.

OCR: *Optical Character Recognition*. Técnica que convierte en texto los caracteres presentes en una imagen.

PaaS: *Platform as a Service*. Modelo en que el proveedor administra además la plataforma de ejecución de la aplicación.

QA: *Quality Assurance*. Ambiente de calidad en que se prueba una solución con datos reales antes de pasarla a producción.

RF: Requerimiento funcional. Lo que debe hacer la solución, con independencia de cómo se implemente.

RPA: *Robotic Process Automation*. Automatización que opera sobre la interfaz gráfica de los sistemas, imitando las acciones de un usuario.

S3: *Simple Storage Service*. Servicio de almacenamiento de objetos de AWS, accesibles por clave.

SaaS: *Software as a Service*. Modelo en que el proveedor entrega una aplicación terminada.

SQS: *Simple Queue Service*. Servicio de colas de mensajes de AWS.

SX: Empresa suministradora de energía eléctrica. Vende la energía al cliente libre y emite su propia factura.

TI: Tecnologías de la información. En esta memoria, el área de Desarrollo y Tecnología de Evol Services.

USD: Dólar estadounidense. Moneda en que se expresan los costos de infraestructura.

INDICE DE CONTENIDOS

DEDICATORIA 2

AGRADECIMIENTOS 3

RESUMEN 4

ABSTRACT 4

GLOSARIO 5

INDICE DE CONTENIDOS 7

INDICE DE FIGURAS 9

INDICE DE TABLAS 10

INTRODUCCIÓN 12

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA 14

1.1 CONTEXTO 14

1.2 ORGANIZACIÓN 15

1.3 SITUACIÓN ACTUAL 18

1.4 ACTORES INVOLUCRADOS 22

1.5 DIMENSIONAMIENTO DEL PROBLEMA 24

1.6 ANÁLISIS E IMPACTO DE NO SOLUCIONAR EL PROBLEMA 28

1.7 COMPETENCIAS EXISTENTES 31

1.8 OBJETIVOS 33

1.9 ALCANCE 33

CAPÍTULO 2: MARCO CONCEPTUAL 35

2.1 ARQUITECTURA DE *SOFTWARE* Y ATRIBUTOS DE CALIDAD 35

2.2 ESTILOS ARQUITECTÓNICOS 37

2.3 COMPUTACIÓN EN LA NUBE 41

2.4 EXTRACCIÓN DE DATOS DESDE DOCUMENTOS 45

2.5 TRABAJOS RELACIONADOS 48

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN 51
3.1 ANTECEDENTE Y REQUERIMIENTOS 51
3.2 SELECCIÓN DE SERVICIOS 56
3.3 ARQUITECTURA PROPUESTA: VISIÓN GENERAL 58
3.4 DECISIONES ARQUITECTÓNICAS 61
3.5 DISEÑO DETALLADO DE LOS SERVICIOS 67
3.6 ASPECTOS TRANVERSALES 71
3.7 ALCANCE DE LA IMPLEMENTACIÓN 73
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN 75
4.1 DISEÑO DE LA VALIDACIÓN 75
4.2 PRECISIÓN DE LA EXTRACCIÓN 76
4.3 ATRIBUTOS DE CALIDAD 78
4.4 COSTO Y ESFUERZO 83
4.5 RESUMEN Y LÍMITES DE LA VALIDACIÓN 85
CAPÍTULO 5: CONCLUSIONES 88
5.1 CONCLUSIONES GENERALES 88
5.2 CUMPLIMIENTO DE OBJETIVOS 89
5.3 TRABAJO FUTURO 91
5.4 PALABRAS FINALES 92
REFERENCIAS BIBLIOGRÁFICAS 93

INDICE DE FIGURAS

Figura 1: Árbol del problema. Fuente: Elaboración propia. 29

Figura 2: Diagrama de contexto de la solución (C4, nivel 1). Fuente: Elaboración propia. . 59

Figura 3: Diagrama de contenedores de la solución (C4, nivel 2). Fuente: Elaboración propia. 61

INDICE DE TABLAS

Tabla 1: Limitaciones de la automatización actual. Fuente: Elaboración propia a partir de datos de Evol. 21

Tabla 2: Actores involucrados en el proceso de gestión de facturas. Fuente: Elaboración propia a partir de datos de Evol. 23

Tabla 3: Conteo de facturas por emisor. Fuente: Elaboración propia a partir de datos de Evol. 25

Tabla 4: Línea base del proceso de gestión de facturas. Fuente: Elaboración propia a partir de datos de Evol. 26

Tabla 5: Relación entre los problemas identificados, los atributos de calidad exigidos y las métricas de validación. Fuente: Elaboración propia. 30

Tabla 6: Categorías de soluciones existentes para la extracción de datos desde facturas. Fuente: Elaboración propia. 31

Tabla 7: Atributos de calidad seleccionados, según ISO/IEC 25010:2023 y Capítulo 1. Fuente: Elaboración propia a partir de ISO/IEC 25010:2023. 37

Tabla 8: Beneficios y costos de una arquitectura de microservicios. Fuente: Elaboración propia a partir de Lewis y Fowler (2014), Newman (2021) y Richards y Ford (2020). 40

Tabla 9: Categorías de servicios utilizadas y el rol que cumplen en este trabajo. Fuente: Elaboración propia. 43

Tabla 10: Comparación de enfoques de extracción de datos desde facturas. Fuente: Elaboración propia a partir de Yashwant et al. (2025), Xu et al. (2020) y Perot et al. (2023). 48

Tabla 11: Trabajos relacionados y su relación con esta memoria. Fuente: Elaboración propia. 49

Tabla 12: Requerimientos funcionales. Fuente: Elaboración propia. 51

Tabla 13: Escenarios de calidad de la solución. Fuente: Elaboración propia a partir de la Tabla 7 y de la línea base de la Tabla 4. 53

Tabla 14: Restricciones del diseño. Fuente: Elaboración propia a partir de datos de Evol. . 55

Tabla 15: Servicio seleccionado para cada componente de la solución. Fuente: Elaboración propia. 56

Tabla 16: Resumen de las decisiones arquitectónicas. Fuente: Elaboración propia. 66

Tabla 17: Reglas del servicio de validación. Fuente: Elaboración propia. 69

Tabla 18: Costo mensual estimado de operación para 800 facturas. Fuente: Elaboración propia a partir de las tarifas publicadas por Amazon Web Services. 72

Tabla 19: Estado de cada componente al cierre de la memoria. Fuente: Elaboración propia. 73

Tabla 20: Conjunto de prueba por emisor. Fuente: Elaboración propia a partir de datos de Evol. 76

Tabla 21: Precisión de la extracción por grupo de campos. Fuente: Elaboración propia. ... 78

Tabla 22: Errores de extracción por tipo en facturas procesadas. Fuente: Elaboración propia. 78

Tabla 23: Respuesta de la solución y del bot ante fallas. Fuente: Elaboración propia a partir de datos de Evol. 79

Tabla 24: Costo y esfuerzo mensual de operación. Fuente: Elaboración propia a partir de datos de Evol. 84

Tabla 25: Resumen de la validación. Fuente: Elaboración propia. 85

INTRODUCCIÓN

Muchos procesos administrativos dependen de información que no llega como datos estructurados, sino contenida en documentos pensados para ser leídos por personas. Las facturas son el caso más común: cada emisor define su propio formato, la disposición de la información y el nombre de cada concepto, de modo que trasladar sus datos a los sistemas internos suele requerir lectura y transcripción manual. Cuando el volumen es alto y los datos alimentan otros procesos, esa transcripción se convierte en un cuello de botella operacional, por el tiempo que consume y por el riesgo de error que introduce. Así ocurre en la gestión de contratos de energía eléctrica para clientes libres, donde verificar mes a mes lo que cobra cada factura es parte central del servicio que se presta.

Este trabajo aborda ese problema en Evol Services, empresa que gestiona los contratos de cerca de 400 puntos de suministro, pertenecientes a unas 360 empresas clientes. Cada mes se reciben alrededor de 800 facturas, emitidas por distribuidoras y suministradoras en 30 formatos distintos, cuyos datos deben cargarse en el sistema interno EMER para elaborar el Reporte de Validación, que contrasta lo cobrado con lo calculado antes de que el cliente pague. La carga, originalmente manual, demandaba unas 60 horas-persona al mes. Una primera automatización, un *bot* basado en reglas programadas para cada formato, redujo ese esfuerzo a cerca de 28 horas-persona, pero falla en alrededor del 30% de las facturas, que siguen cargándose a mano, y más de un tercio del esfuerzo restante corresponde a mantenerlo. Además, sus extractores fallan cuando un emisor cambia su formato, todo cambio exige desplegar el proceso completo y no existe una vista del estado de cada factura.

Para superar esas limitaciones, este trabajo propone el diseño, la implementación y la validación de una arquitectura de microservicios en la nube, construida sobre la infraestructura que la empresa ya utiliza, que automatiza la extracción, la validación y la carga de los datos de las facturas y reemplaza progresivamente al *bot*. El alcance comprende las etapas de revisión y carga, desde que la factura está disponible en el repositorio documental de la empresa hasta su registro en EMER, la obtención de las facturas y las facturas compuestas por más de un documento quedan fuera y se declaran como trabajo futuro.

El trabajo se desarrolló en etapas sucesivas. Primero se analizó el proceso actual y se caracterizaron sus dos líneas base, el procedimiento manual y el *bot*. A partir de ese análisis se definieron los requerimientos de la solución y los atributos de calidad prioritarios, expresados como escenarios con metas medibles. Luego se seleccionaron los servicios en la nube para cada componente y se diseñó la arquitectura, documentada mediante el modelo C4 y registros de decisión arquitectónica que justifican cada elección frente a sus alternativas. Después se implementaron los servicios, incluido el seguimiento del estado de cada factura. Por último, se validó la solución en el ambiente de calidad (QA) de la empresa

con facturas reales, comparándola con el procedimiento manual y con el *bot* en precisión de extracción, tiempo de procesamiento, esfuerzo de mantención y costo operacional.

El documento se organiza en cinco capítulos. En el Capítulo 1 se presentan el contexto y la organización, se describe la situación actual, se dimensiona el problema y se plantean los objetivos y el alcance del trabajo. En el Capítulo 2 se exponen los conceptos que sustentan el diseño: arquitectura de software, estilos arquitectónicos, computación en la nube y extracción de datos desde documentos, junto con trabajos relacionados. En el Capítulo 3 se definen los requerimientos y escenarios de calidad, se seleccionan los servicios, se presentan la arquitectura propuesta y sus decisiones, y se detalla el diseño de cada servicio. En el Capítulo 4 se describe el protocolo de validación y se presentan los resultados obtenidos frente a las metas definidas. Finalmente, en el Capítulo 5 se exponen las conclusiones, el cumplimiento de los objetivos y las líneas de trabajo futuro.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 CONTEXTO

Un gran número de procesos administrativos depende de información que no está en bases de datos ni mediante plataformas automatizadas, sino que llega en documentos diseñados para ser leídos por personas. Las facturas son el caso más extendido, cada emisor decide cómo estructurar la información, qué nombre dar a cada concepto y en qué disposición presentarlo, de modo que quien recibe documentos de múltiples emisores debe interpretar tantos formatos distintos como proveedores tenga. El contenido es conceptualmente equivalente, pero su representación explícita no lo es, y esa diferencia es la que impide tratarlos como datos estructurados.

La consecuencia habitual es que la información se traslada a los sistemas internos mediante transcripción manual, con el costo y el riesgo que ello implica. El costo crece de forma proporcional al volumen, porque cada documento tiene el mismo trabajo de lectura, interpretación e ingreso independiente del contenido, y no disminuye con la práctica más allá de cierto punto. El riesgo viene del error, la investigación sobre digitación manual de datos muestra tasas de error que, aun siendo bajas por concepto, se vuelven significativas al multiplicarse por el volumen procesado[1], y que resultan especialmente difíciles de detectar cuando el dato transcrito es aceptable pero incorrecto, ya que no incumple ninguna regla evidente y solo se descubre al compararlo con una fuente independiente. Cuantificar ese costo y ese riesgo en un caso concreto es, precisamente, uno de los propósitos del análisis que se desarrolla en este capítulo.

En el sector eléctrico estas condiciones se intensifican. Una factura de suministro no contiene un único monto, sino un desglose de cargos (energía, potencia, transmisión, cargos regulados y otros ítems que varían según el decreto tarifario vigente) cuya composición cambia entre emisores y entre periodos. Verificar que ese desglose corresponde a lo contratado es una actividad mensual, obligatoria y con impacto económico directo: un cargo mal interpretado o un valor mal transcrito se traduce en un pago incorrecto. Para las empresas que gestionan contratos de suministro por cuenta de terceros, esta verificación no es una tarea administrativa más, sino el núcleo del servicio que prestan, lo que convierte la calidad y la oportunidad de esos datos en una condición de su operación.

[1] Mays & Mathias (2019); Panko (2008). Ambos estudios provienen de ambientes distintos al de la facturación eléctrica (el análisis clínico y las planillas de cálculo, respectivamente), se citan por su caracterización de la tasa de error en la transcripción manual de datos, no como referencia del sector.

1.2 ORGANIZACIÓN

1.2.1 MERCADO ELÉCTRICO Y LOS CLIENTES LIBRE

El suministro de energía eléctrica en Chile involucra tres partes conectadas: la generación, la transmisión y la distribución. Desde la perspectiva de un usuario final, esta cadena se traduce en dos relaciones comerciales distintas. Por una parte, la empresa suministradora (SX) le vende la energía que consume. Por otra, la empresa distribuidora (DX) opera la red que la transporta hasta sus instalaciones y le cobra por ese uso, además de los cargos regulados asociados al sistema. En consecuencia, cada punto de suministro (llamado principalmente, instalación) recibe mensualmente dos facturas de emisores distintos, cada una con su propia estructura de cargos.

La legislación chilena distingue dos regímenes de precios para estos usuarios. Los clientes regulados pagan tarifas fijadas por la autoridad, mientras que los clientes libres pactan el precio de su suministro mediante negociación directa con el proveedor. La pertenencia a uno u otro régimen depende de la potencia conectada de la instalación. Los usuarios finales cuya potencia conectada supera los 5.000 kW quedan sujetos a precio libre, por aplicación de la Ley General de Servicios Eléctricos (Ministerio de Economía, Fomento y Reconstrucción, 2007). Aquellos cuya potencia conectada se sitúa entre 300 y 5.000 kW son considerados clientes elegibles y pueden optar entre ambos regímenes, debiendo permanecer al menos cuatro años en el que escojan y comunicar con doce meses de anticipación cualquier cambio (Comisión de Minería y Energía del Senado, 2019). Este límite inferior se redujo de 500 a 300 kW mediante la Resolución Exenta Nro. 58 de 2024 (Ministerio de Energía, 2024)[2], lo que amplió el universo de usuarios que pueden acceder al mercado libre.

Negociar directamente el suministro ofrece a estos usuarios la posibilidad de obtener condiciones más convenientes que las tarifas reguladas, pero les traslada una responsabilidad que antes no tenían: comparar ofertas, administrar contratos de largo plazo, verificar que los precios pactados se apliquen correctamente y controlar cargos que varían mes a mes según decretos tarifarios y condiciones del sistema. Esa complejidad explica la existencia de empresas especializadas en gestionar estos contratos por cuenta del cliente.

1.2.2 EVOL SERVICES

Evol Services SpA es una empresa chilena dedicada a la gestión energética y la eficiencia energética, forma parte de EVOL, marca que agrupa además otras unidades de negocio orientadas a la comercialización de energía y a la generación solar. La marca se unificó en octubre de 2023 como filial de LipiAndes, del grupo Lipigas, si bien su trayectoria en la administración de contratos eléctricos en el mercado libre chileno se remonta aproximadamente a 2016.

[2] Ministerio de Energía (2024). Resolución Exenta N° 58. Posteriormente, la Resolución Exenta N° 13, de febrero de 2025, precisó el criterio con que se determina la potencia conectada, sin modificar el umbral de 300 kW.

Dentro de su variada cartera de servicios, el presente trabajo se sitúa en el de gestión de contratos de energía eléctrica. Bajo este, Evol busca y administra por cuenta de sus clientes los contratos más convenientes, tanto con suministradoras como con distribuidoras, y controla mensualmente que lo cobrado corresponda a lo pactado. Su principal modelo de ingresos para este servicio se basa en cobrar un porcentaje del ahorro generado al cliente, entendido como la diferencia entre lo que habría pagado bajo el régimen regulado y lo que efectivamente paga bajo el régimen libre gestionado por Evol. El negocio está orientado a usuarios de alto consumo, aunque no excluye a clientes de menor tamaño.

La unidad de gestión no es la empresa cliente, sino el punto de suministro. Una misma organización con varias instalaciones (sucursales, plantas o locales) mantiene una medición y una facturación independientes por cada una de ellas, y cada una se administra por separado a lo largo de todo el proceso. Actualmente Evol gestiona cerca de 400 instalaciones, correspondientes a aproximadamente 360 empresas clientes[3]. Puesto que cada punto recibe una factura de su suministradora y otra de su distribuidora, el volumen mensual asciende a cerca de 800 facturas.

1.2.3 REPORTE DE GESTIÓN Y VALIDACIÓN

Para determinar cuánto cobrar mensualmente a cada cliente, Evol elabora dos reportes complementarios.

El **Reporte de Gestión** corresponde al cálculo completo de lo que el cliente debería pagar y su desglose por cargos, considerando consumo, tarifas y decretos vigentes, tanto en el mercado regulado como en el libre. La diferencia entre ambos escenarios refleja el ahorro del mes, sobre el cual se calcula la comisión de Evol. Este reporte se elabora siempre primero y a principio de mes, gracias a la dinámica misma que tienen sus insumos (mediciones y tarifas) donde distribuidoras responden a las estrictas condiciones de plazos que tiene el Coordinador Eléctrico Nacional (CEN).

El **Reporte de Validación** contrasta ese cálculo con la realidad. Como los valores del Reporte de Gestión son calculados, deben compararse cargo a cargo con lo que la suministradora y la distribuidora detallan y cobran efectivamente en sus facturas. Este contraste ocurre antes de que el cliente pague y es el mecanismo con que Evol detecta cargos indebidos, disparidades en la medición del consumo o precios distintos a los pactados. El reporte expresa cada diferencia como un desvío porcentual entre el valor calculado y el facturado, y lo clasifica según su magnitud (moderado o incidencia) de modo que sirve simultáneamente como entregable al cliente y como alerta para el analista (correo consolidado con alertas y reporte). Dado que el valor calculado se considera confiable, un desvío significativo apunta a un problema en la factura o en el proceso intermedio de carga.

Ambos reportes llegan al cliente y a los analistas de Evol. A diferencia del Reporte de Gestión, el de Validación no tiene una fecha fija: depende de cuándo llegue la factura de

[3] Todas las cifras operacionales de este capítulo provienen de información entregada por el Coordinador del equipo de Gestión y Control de Procesos.

cada emisor. Cuando una factura llega demasiado tarde, se cobra directamente lo estimado en el Reporte de Gestión y la diferencia se reliquida el mes siguiente, lo que traslada al periodo posterior tanto el ajuste como su verificación.

Es relevante destacar cómo se encadenan estos pasos dentro del sistema interno de Evol: al cargarse una factura, se gatilla automáticamente la compilación de su Reporte de Validación y su posterior envío al cliente. Es decir, todas las etapas posteriores a la disponibilidad de los datos ya están automatizadas. Los datos contenidos en la factura son, por tanto, el insumo esencial de la validación de cobros, y la operación de incorporarlos al sistema es el único paso de la cadena que no lo está.

1.2.4 EL EQUIPO GCP Y CONTEXTO TECNOLÓGICO

El proceso anteriormente descrito se ejecuta en el equipo de Gestión y Control de Procesos (GCP), responsable de administrar los insumos de los reportes (facturas, mediciones, tarifas y contratos), de disponerlos en el sistema interno y de elaborar los reportes mismos. El equipo lo integran ocho personas, entre ellas un coordinador, un analista de gestión de información energética, que es el responsable directo de los insumos de facturación y cuenta con dos alumnos en práctica a su cargo, un analista de aseguramiento de calidad (*Quality Assurance, QA*), una diseñadora de experiencia de usuario (*User Experience/User Interface, UX/UI*) y un analista de procesos encargado de levantar los requerimientos de las distintas áreas. GCP opera como cliente interno del área de Desarrollo y Tecnología (TI), integrada por otras siete personas, que mantiene los sistemas de la empresa y desarrolla sus nuevas funcionalidades.

El sistema donde se registran y procesan estos datos es EMER, un desarrollo propio de Evol, originalmente externalizado y luego asumido por el área interna. Su núcleo es una aplicación monolítica construida en Java con Spring Boot sobre una base de datos relacional PostgreSQL, donde se incorporó posteriormente una interfaz en React. Desde hace aproximadamente dos años, los desarrollos nuevos y las funcionalidades que se migran desde el monolito se construyen en NestTS y se despliegan sobre servicios administrados de AWS (principalmente Lambdas), infraestructura que la empresa ya utiliza y dentro de la cual convive con el componente Java, este último alojado junto a la interfaz (*front*) en una instancia EC2 (servidor virtual en la nube) del mismo AWS. La documentación operativa de la empresa, incluidas las facturas recibidas, se almacena en SharePoint, dentro de la suite Microsoft 365. Este contexto tecnológico no indica por sí mismo una solución, pero delimita el entorno en que cualquier propuesta debe crearse y se retoma como restricción de diseño en el Capítulo 3

1.3 SITUACIÓN ACTUAL

La gestión de facturas en Evol Services consta de tres etapas: obtención, revisión y carga. La obtención se realiza a través de los portales web de suministradoras y distribuidoras o por correo electrónico, y los documentos (formato PDF en su mayoría), se almacenan en el repositorio SharePoint organizados por tipo y emisor. Esta etapa no forma parte del presente trabajo. Las dos etapas siguientes son el objeto de esta memoria y hoy conviven en dos modalidades: una automatización que intenta procesar la totalidad de las facturas y lo consigue en cerca del 70% de los casos, y un procedimiento manual que absorbe el 30% restante. Esta sección describe primero el procedimiento manual, que es el original y el que define el costo del proceso, y luego la automatización actual y sus limitaciones.

1.3.1 PROCESO DE REVISIÓN Y CARGA MANUAL

Una vez disponible la factura, el procedimiento manual comprende los siguientes pasos:

Creación del periodo. En EMER se crea el mes correspondiente, lo que genera la lista de instalaciones con sus valores en cero. Actualmente, la primera carga de cada mes suele crear el periodo de forma automática, por lo que este paso rara vez se ejecuta a mano.

Revisión. Se abre la factura y se identifican los cargos relevantes, registrando para cada uno su concepto, consumo y costo, junto con los metadatos del documento: número de folio, nombre del titular, fecha de emisión, periodo facturado y total neto. Las facturas de distribuidoras contienen habitualmente unos cinco cargos de interés para la validación, mientras que las de suministradoras presentan entre cinco y quince, según el emisor y según si el periodo incluye reliquidaciones. La extensión de los documentos leídos también varía, hay facturas de una página y otras de hasta siete.

Categorización. Cada cargo identificado debe asignarse a un concepto del sistema, porque un mismo ítem aparece con nombres distintos según el emisor. Por ejemplo, "Energía activa" y "Energía consumida" corresponden ambos al concepto "Energía". EMER mantiene para ello un diccionario administrado por el equipo GCP, en el que cada concepto general agrupa la lista de denominaciones con que puede aparecer en los distintos formatos. Este diccionario contempla seis conceptos generales para distribuidoras y cerca de treinta y dos para suministradoras (se pueden quitar o crear cargos según corresponda), diferencia que refleja la mayor diversidad de ítems que estas últimas incorporan a sus cobros. La configuración de ese diccionario determina directamente cómo se registra cada cargo.

Carga. Los valores se ingresan en los formularios de EMER, que los almacena en su base de datos relacional. Esta carga es la que gatilla, de forma automática, la compilación del Reporte de Validación correspondiente.

Verificación. No existe un procedimiento sistemático de revisión del detalle cargado. El control efectivo es binario (se verifica que la factura esté o no cargada) y la detección de errores queda entregada al propio Reporte de Validación, que alerta cuando un valor se

desvía de lo calculado. En la práctica, cada generación de reportes arroja alertas, y no se registran meses enteros sin ellas.

A esta secuencia se suman casos de entrada que no se ajustan al patrón de un sólo PDF por factura, llámese casos borde o especiales: un emisor de los tantos, acompaña su factura con una proforma en formato de planilla, que contiene el detalle de los cargos reliquidados y debe cargarse en conjunto con el PDF; algunos emisores emiten dos documentos por periodo, el segundo archivo con un único cargo aplicable solo a ese mes; y otro emisor entrega sus archivos protegidos con contraseña, que debe retirarse antes de procesarlos.

Antes de la automatización, esta tarea recaía generalmente en dos personas del equipo GCP, una dedicada a las facturas de distribuidoras y otra a las de suministradoras. Cada una destinaba alrededor de 30 horas mensuales, distribuidas a lo largo de unas dos semanas y en paralelo con sus demás funciones, lo que totaliza aproximadamente 60 horas-persona al mes. Ese reparto responde a que las facturas no llegan todas a la vez ni en una fecha fija, sino de manera dispersa, en general entre la segunda y la tercera semana del mes, por lo que la carga se procesa a medida que los documentos van estando disponibles. En la actualidad, la porción que sigue siendo manual suele quedar a cargo de un analista junto a un alumno en práctica, o únicamente de este último, lo que resulta relevante porque quienes ejecutan la tarea con menor experiencia enfrentan una dificultad adicional: interpretar correctamente treinta formatos distintos y asignar cada cargo al concepto que corresponde exige un conocimiento que se adquiere con el tiempo y que se pierde con cada rotación.

1.3.2 AUTOMATIZACIÓN ACTUAL: EL *BOT*

Para aliviar esta carga operativa se implementó una primera automatización, en adelante el *bot*, desarrollada internamente (por el autor de este documento) y hoy en producción. Su construcción respondió a una urgencia operativa más que a un diseño arquitectónico formal: comenzó cubriendo los dos formatos de distribuidoras que concentran cerca de 300 facturas mensuales, y los formatos restantes se fueron incorporando de manera incremental a lo largo de los meses siguientes, hasta disponer hoy de un extractor para cada uno de los treinta formatos vigentes. El *bot* demostró que automatizar el proceso es factible y redujo de manera significativa el esfuerzo del equipo, razón por la cual constituye, junto con el procedimiento manual, una de las dos líneas base de esta memoria.

Su funcionamiento es el siguiente. Cada ejecución del bot recibe dos parámetros: una carpeta de SharePoint y el formato que debe aplicar. Existe una carpeta reservada por formato (emisor), y una tarea programada mediante *cron* (en servidor EC2 de AWS) encadena treinta ejecuciones secuenciales, una por formato, a las 20:00 de cada día, el orden secuencial evita sobrecargar el servidor. Para cada carpeta con contenido, el *bot* descarga los documentos, los procesa uno a uno extrayendo los datos mediante reglas programadas específicamente para ese formato, y envía los valores obtenidos al sistema EMER, invocando su interfaz de programación (*Application Programming Interface*, *API*) con una estructura que incluye tanto los metadatos de la factura como la lista de cargos con su

concepto, consumo y costo. Las facturas procesadas correctamente se trasladan a su carpeta definitiva, ordenada por año, mes y emisor, y las que fallan se depositan en una carpeta de errores. Si una factura falla, el proceso continúa con la siguiente: un error individual no detiene el lote.

Dos aspectos de esta operación que deben tener atención porque no son evidentes. El primero es que el *bot* no lee la carpeta donde llegan las facturas, sino una carpeta específica por formato en la que el analista debe depositar las facturas a procesar, el traslado de cada documento hasta esa ubicación sigue siendo una tarea humana, contabilizada dentro del esfuerzo manual del proceso. El segundo es que el *bot* registra únicamente los errores, en un reporte que se envía por correo al analista, sin dejar constancia de las cargas exitosas. En consecuencia, la detección de un problema ocurre por dos vías indirectas: el correo de errores o la ausencia del Reporte de Validación que se esperaba recibir. Reintentar una factura fallida es también una acción manual, que consiste en devolver el archivo a la carpeta del formato.

Con esta configuración, el *bot* procesa correctamente alrededor del 70% de las facturas mensuales, a razón de unos 30 segundos por documento. El 30% restante falla y debe revisarse y cargarse a mano. Conviene subrayar que este 30% no corresponde a formatos sin cobertura, los treinta formatos vigentes cuentan con extractor, y aun así una de cada tres facturas no logra procesarse. La explicación está en que las reglas de extracción, basadas en la posición de los datos o en patrones de texto, no toleran la variabilidad que existe dentro de un mismo formato, cargos que aparecen o desaparecen según el cliente, tablas de largo variable, periodos con reliquidaciones, ajustes menores de diseño y casos borde que no se consideraron. Ordenadas de mayor a menor frecuencia, las causas de falla son: el emisor modificó el formato de su factura; un valor se lee de forma incorrecta, de modo que la conversión al tipo de dato esperado interrumpe el procesamiento de esa factura; un campo obligatorio no se encuentra, en particular las fechas, necesarias para asociar la factura a un periodo; un concepto no figura en el diccionario y no se carga; el archivo resulta ilegible; y, de forma marginal, una falla del propio sistema receptor (EMER). Estas fallas se concentran de manera notoria en las facturas de suministradoras, cuyos formatos cambian con mayor frecuencia y presentan más casos particulares que los de distribuidoras.

Lo relevante es que llevar un extractor basado en reglas al 100% de acierto no es viable: cada caso borde exige una regla adicional, y cada regla adicional aumenta la superficie que puede romperse ante el siguiente cambio. La magnitud de ese acoplamiento es visible en el propio código, de las aproximadamente 6.000 líneas del proyecto, cerca de 4.400 (un 73%) corresponden a los treinta extractores, es decir, a lógica específica por formato que debe mantenerse una por una. El enfoque llegó a su límite, y no se resuelve programando más extractores.

De esta operación se desprenden las limitaciones que motivan el rediseño, identificadas (ID respectivo) y retomadas en los capítulos siguientes.

Tabla 1: Limitaciones de la automatización actual. Fuente: Elaboración propia a partir de datos de Evol.

ID	Limitación	Evidencia
L1	Fragilidad de la extracción. Las reglas no toleran la variabilidad interna de un formato y dejan de funcionar cuando el emisor lo modifica.	~30% de las facturas falla pese a existir extractor para los 31 formatos, 10 interrupciones por cambio de formato en tres meses, reparación de 4 a 10 horas por incidencia.
L2	Costo de extender la cobertura. Los 31 formatos son los que la empresa maneja hoy, pero el universo de emisores es mayor, cada emisor nuevo exige programar y validar un extractor adicional.	~25 horas por formato nuevo, equivalentes a unos tres días de jornada, incluidas las pruebas.
L3	Acoplamiento monolítico. Todas las etapas residen en un único proceso, alojado además en el mismo servidor que el EMER (EC2). Una factura fallida no detiene el lote, pero la caída del proceso o del servidor deja el día sin procesar.	Cuando procesos saturan el servidor y el EMER deja de responder, falla el lote completo y requiere reiniciar la instancia. La actualización de una biblioteca que se usaba para el acceso a SharePoint, bloqueó en su momento toda operación hasta sustituirla.
L4	Latencia por ejecución programada. Al ejecutarse una vez al día, una factura puede tardar hasta 24 horas en estar disponible, con mayor impacto en los días de mayor recepción.	Ejecución diaria a las 20:00, las facturas llegan durante la jornada laboral.
L5	Infraestructura fija y punto único de falla. Un servidor permanentemente encendido, sin recuperación automática, con costo constante aunque la carga se concentre en pocos días del mes.	Costo de la instancia entre 120 y 130 dólares mensuales, compartido con otros componentes del sistema.
L6	Visibilidad limitada. Solo se registran los errores, de las cargas exitosas no queda constancia. El equipo GCP carece de una vista del estado de cada factura y se entera de los problemas por el correo de errores o por la ausencia del reporte esperado. Reprocesar es una acción manual.	Reporte de errores por correo, logs de ejecución accesibles solo al equipo TI.

L7	Mantenibilidad. El 73% del código es lógica específica por formato, sin separación de responsabilidades ni pruebas automatizadas, y el conocimiento se concentra en una persona.	~6.000 líneas en total, ~4.400 en extractores, el mantenimiento lo realiza principalmente un desarrollador.

1.3.3 SÍNTESIS

La experiencia con la automatización actual confirma la complejidad del problema y, a la vez, delimita con precisión su naturaleza. El *bot* demostró que extraer y cargar los datos de forma automática es factible, y redujo de manera sustantiva el esfuerzo del equipo. Pero también evidenció que un enfoque basado en reglas específicas por formato es frágil ante los cambios de las facturas, costoso de extender a nuevos emisores y difícil de operar cuando todas las etapas dependen de un único proceso (monolito). A ello se suma que la operación carece de visibilidad, no se sabe qué ocurrió con cada factura, salvo cuando algo falla explícitamente.

Por lo tanto, el desafío no es únicamente en automatizar una tarea, sino diseñar una solución que siga funcionando cuando los formatos cambien, que permita incorporar nuevos emisores con bajo esfuerzo, que aísle las fallas de cada etapa y que entregue visibilidad del estado de cada factura. Estas cuatro exigencias, derivadas directamente de las limitaciones observadas, constituyen los atributos de calidad que guían el diseño propuesto en el Capítulo 3 y las dimensiones sobre las que se mide su validación en el Capítulo 4.

1.4 ACTORES INVOLUCRADOS

El proceso de revisión y carga de facturas afecta de manera directa a quienes lo ejecutan y lo mantienen, y de manera indirecta a quienes dependen de sus resultados. La Tabla 2 resume los actores identificados, su rol en el proceso y la forma en que las dificultades actuales los alcanzan.

Tabla 2: Actores involucrados en el proceso de gestión de facturas. Fuente: Elaboración propia a partir de datos de Evol.

Actor	**Rol**	**Efecto**	**Tipo**
Equipo de Gestión y Control de Procesos (GCP)	Administra los insumos de los reportes (facturas, mediciones, tarifas y contratos), los dispone en EMER. El analista de gestión de información energética es el responsable directo de la facturación.	Ejecuta la porción manual de la carga y alimenta las carpetas que procesa la automatización. El tiempo destinado a estas tareas se resta a la elaboración y el análisis de reportes, que es su labor de mayor valor.	Directo
Alumnos en práctica del área GCP	Con frecuencia asumen la revisión y carga manual de las facturas que la automatización no logra procesar, solos o junto a un analista.	Su menor experiencia aumenta el riesgo de error y los tiempos de ejecución, en una tarea que exige conocer treinta formatos y asignar correctamente cada cargo. La rotación periódica obliga a repetir esa curva de aprendizaje.	Directo
Equipo de Desarrollo y Tecnología (TI)	Mantiene EMER y la automatización actual, y desarrolla las nuevas funcionalidades de la empresa.	Debe reprogramar y validar el extractor correspondiente cada vez que un emisor modifica su formato, desarrollar uno nuevo por cada emisor que se incorpora y atender los reprocesos que el equipo GCP no puede resolver por sí mismo. Cada incidencia desvía a un desarrollador de los proyectos en curso.	Directo
Gerencia de Evol Services	Dirige la operación de la empresa y evalúa el desempeño de sus equipos	El tiempo que GCP destina a la carga manual retrasa la emisión de los reportes y, con ello, el flujo normal de la operación, los datos cargados tardíamente o con errores introducen inexactitudes en el ingreso del mes, que se calcula mayoritariamente a partir de emitidos los Reportes de Validación.	Directo

Clientes de Evol Services	Clientes libres que contratan la asesoría y pagan mensualmente las facturas de su suministradora y su distribuidora (considerando servicio específico de estudio).	Si los datos se cargan con errores o tardan en estar disponibles, el Reporte de Validación puede no detectar a tiempo cobros indebidos, diferencias de medición o precios distintos a los contratados, y el cliente podría pagar montos incorrectos. A la inversa, un dato mal cargado genera un desvío aparente y una alerta que llega al cliente sin corresponder a un problema real de la factura.	Indirecto

1.5 DIMENSIONAMIENTO DEL PROBLEMA

Esta sección traduce la situación descrita a cifras, con el doble propósito de dimensionar el problema y establecer la línea base contra la cual se compara la solución propuesta en el Capítulo 4. Son dos líneas base y no una: el procedimiento manual original y la automatización actual.

Volumen y variedad. Evol gestiona cerca de 400 instalaciones, cada uno de los cuales recibe mensualmente una factura de su suministradora y otra de su distribuidora, lo que da un total aproximado de 800 facturas al mes. Estas provienen de 14 distribuidoras y 17 suministradoras, cada una con su propio formato, de modo que el proceso debe tratar con 31 formatos distintos. El volumen se reparte de manera muy desigual entre ellos: dos distribuidoras concentran alrededor de 330 facturas mensuales, mientras que en las suministradoras el reparto es considerablemente más disperso, con un emisor mayor que ronda las 100 facturas y varios que apenas alcanzan unas pocas decenas. Esta concentración es relevante porque significa que una parte importante del volumen se cubre con pocos formatos, mientras que una gran parte de emisores menores exige un esfuerzo de desarrollo y mantención que no guarda proporción con las facturas que aporta.

Tabla 3: Conteo de facturas por emisor. Fuente: Elaboración propia a partir de datos de Evol.

Distribuidor	Facturas	Suministrador	Facturas	Suministrador	Facturas
DX-1	187	**SX-1**	107	**SX-15**	5
DX-2	149	**SX-2**	66	**SX-16**	4
DX-3	15	**SX-3**	62	**SX-17**	3
DX-4	13	**SX-4**	38		
DX-5	9	**SX-5**	22		
DX-6	7	**SX-6**	15		
DX-7	3	**SX-7**	12		
DX-8	2	**SX-8**	12		
DX-9	2	**SX-9**	11		
DX-10	2	**SX-10**	9		
DX-11	2	**SX-11**	9		
DX-12	1	**SX-12**	7		
DX-13	1	**SX-13**	6		
DX-14	1	**SX-14**	6		

Esfuerzo. Antes de la automatización, el proceso consumía alrededor de 60 horas-persona al mes. En la actualidad, la carga manual del 30% que la automatización no logra procesar (incluido el traslado de los documentos a las carpetas que esta lee) demanda unas 18 horas-persona al mes, a las que se suman entre 10 y 15 horas-persona mensuales de mantención y soporte de la propia automatización, según la cantidad de incidencias del periodo. Tomando el extremo inferior de ese rango, el esfuerzo total asciende a aproximadamente 28 horas-persona al mes.

Lectura de las cifras. La automatización redujo el esfuerzo en más de la mitad, lo que confirma que el camino es correcto. Sin embargo, dos observaciones puntualizan ese resultado. La primera es que más de un tercio del esfuerzo restante ya no consiste en cargar facturas, sino en mantener la herramienta que las carga, un costo que antes no existía y que crece con cada formato que cambia o se incorpora. La segunda es que ese 30% de falla no se explica por falta de cobertura, puesto que los treinta formatos vigentes disponen de extractor. Ampliar la cobertura, por tanto, no resolvería el problema, el límite está en el enfoque.

Atención periódica. Las facturas no llegan en una fecha fija, sino de forma dispersa, concentrándose entre la segunda y la tercera semana del mes. El proceso debe absorber ese *peak* con una capacidad que permanece constante durante todo el mes, y es también esa ventana la que define el escenario de carga sobre el que se diseña y se prueba la solución en los capítulos siguientes.

Calidad de los datos. No existe hoy un registro sistemático de errores de carga, porque la automatización solo deja constancia de sus fallas y el proceso manual carece de una verificación posterior del detalle ingresado. La única señal disponible son las alertas de desvío del Reporte de Validación, que no distinguen entre un cobro efectivamente irregular y un dato mal cargado.

Tabla 4: Línea base del proceso de gestión de facturas. Fuente: Elaboración propia a partir de datos de Evol.

Métrica	**Proceso Manual**	**Automatización Actual**	**Fuente**
Facturas procesadas al mes.	~800	~800	Equipo GCP
Emisores y formatos	31 (14 DX, 17 SX)	31	Equipo GCP
Formatos con extractor disponible	-	31	Automatización
Facturas procesadas correctamente de forma automática	-	~70%	Automatización y Equipo GCP
Facturas que requieren carga manual	100%	~30%	Automatización y Equipo GCP
Horas-persona al mes en revisión y carga	60 (entre 2, ~30 cada una)	18	Equipo GCP

Horas-persona al mes en soporte y mantención	-	10 a 15	Equipo TI
Esfuerzo total (horas-persona/mes)	60	28 a 33	Cálculo
Tiempo para procesar factura	~4,5 minutos	~30 segundos	Cálculo y automatización
Duración del ciclo mensual	~2 semanas calendario	~2 semanas calendario	Equipo GCP
Latencia entre llegada de factura y carga	~4,5 minutos hasta días (dependía de disponibilidad del analista)	Hasta 24 horas	Equipo GCP y Automatización (*cron*)
Interrupciones por cambio de formato	-	10 en 3 meses (23 *commits*)	Repositorio de código
Esfuerzo por nuevo formato	-	~25 horas	Estimación equipo TI (para desarrollador interiorizado)
Esfuerzo por reparar formato	-	4 a 10 horas	Estimación equipo TI (para desarrollador interiorizado)
Costo de infraestructura.	-	120-130 USD al mes (en conjunto con otros proyectos en EC2)	Equipo TI
Tasa de error	~5%	30%	-

1.6 ANÁLISIS E IMPACTO DE NO SOLUCIONAR EL PROBLEMA

Más allá de tratarse de un proceso repetitivo y con alto uso de recursos humanos, el diagnóstico anterior permite identificar seis problemas, que se enuncian a continuación con un identificador que se mantiene en los capítulos siguientes:

P1. Falta de valor agregado. La lectura y carga de facturas no aporta valor por sí misma, más allá de preparar los datos para su uso en otros procesos. El tiempo que consume se resta a la elaboración y el análisis de reportes.

P2. Alta probabilidad de errores humanos. La transcripción manual de los valores y la asignación de cada cargo a su concepto son tareas propensas a error, y ese error afecta directamente la calidad de la información con que se valida el cobro.

P3. Variabilidad de los formatos. Cada emisor presenta la información con una estructura, una distribución y una denominación de conceptos propias, y esos formatos cambian en el tiempo sin aviso previo. Ello dificulta tanto la revisión manual, que exige conocer cada formato, como su automatización mediante reglas fijas.

P4. Fragilidad y costo de mantención de la automatización actual. La automatización depende de reglas específicas por formato: cuando un emisor modifica su factura, el extractor correspondiente deja de funcionar hasta ser reprogramado, e incorporar un formato nuevo exige desarrollar uno adicional.

P5. Escalabilidad limitada. A medida que aumentan los clientes, los emisores y los formatos, el esfuerzo crece de forma lineal: más facturas que revisar manualmente y más extractores que programar y mantener.

P6. Latencia y baja visibilidad del proceso. Los datos pueden tardar hasta 24 horas en estar disponibles, y el equipo GCP no dispone de una vista del estado de cada factura, se entera de los problemas cuando ya ocurrieron.

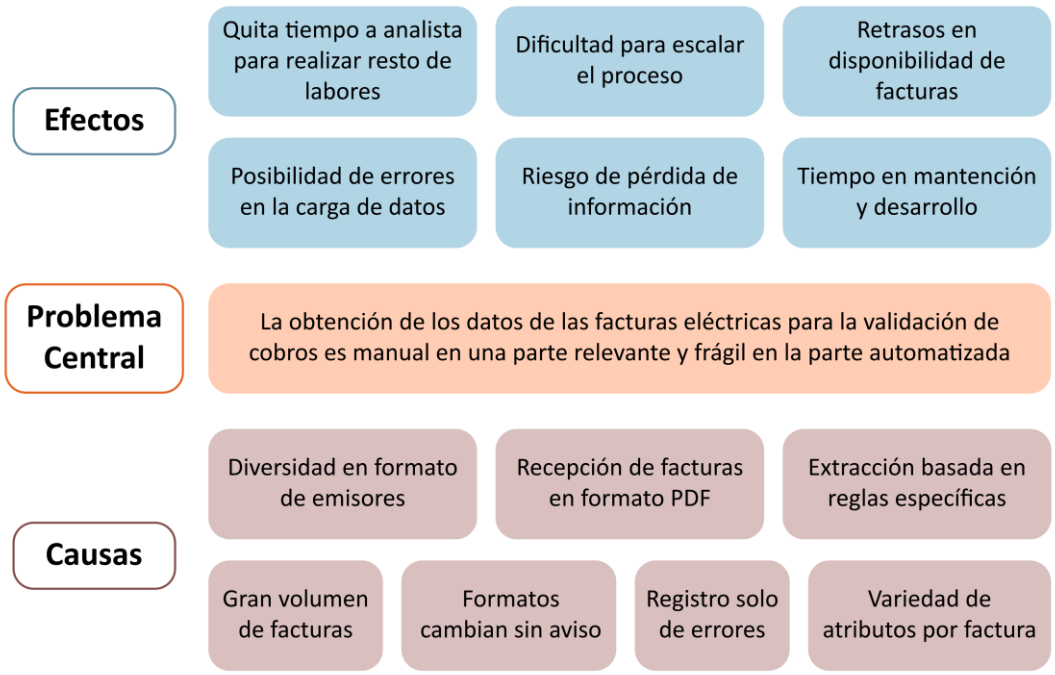


Figura 1: Árbol del problema. Fuente: Elaboración propia.

1.6.2 CONSECUENCIAS DE NO RESOLVER EL PROBLEMA

De mantenerse la situación actual, las consecuencias se proyectan sobre las mismas dimensiones en las que se espera que impacte la solución, lo que permite formular cada una como el escenario contrario al deseado.

Eficiencia operativa: Se mantienen las 18 horas-persona mensuales de carga manual y la disponibilidad tardía de los datos, en desmedro de la elaboración y el análisis de reportes.

Calidad de los datos: Persiste el riesgo de errores de transcripción y de categorización sobre el insumo principal de la validación de cobros, sin un mecanismo que los distinga de un desvío real.

Escalabilidad: Cada punto de suministro nuevo agrega dos facturas mensuales y cada emisor nuevo, unas 25 horas de desarrollo. La rebaja del umbral de potencia conectada a 300 kW amplía el mercado potencial y, con ello, la presión sobre el proceso.

Mantenibilidad: El esfuerzo de mantención crece con cada formato que cambia o se incorpora, y el conocimiento necesario para sostenerlo sigue concentrado en una persona.

Trazabilidad: El equipo GCP continúa sin visibilidad del estado de cada factura, y el reproceso de las fallas recurrentes sigue dependiendo del área de Desarrollo.

Costos: El costo por factura no disminuye: se mantiene el esfuerzo humano, el costo fijo de infraestructura y el costo creciente de mantener la automatización.

En síntesis, el costo del problema no es estático, crece con los clientes, instalaciones y con los formatos, y hoy incluye mantener la propia herramienta creada para resolverlo.

1.6.3 TRAZABILIDAD DISEÑO Y VALIDACIÓN

Los problemas del proceso (P1-P6) y las limitaciones de la automatización actual (L1-L7) se recogen de forma explícita en el diseño y en la validación de la solución, según se resume a continuación. Esta correspondencia se desarrolla como requerimientos y escenarios de calidad en el Capítulo 3 y se verifica mediante las métricas del Capítulo 4.

Tabla 5: Relación entre los problemas identificados, los atributos de calidad exigidos y las métricas de validación. Fuente: Elaboración propia.

Problema	**Limitaciones asociadas**	**Atributo de calidad exigido**	**Métrica de validación**
P1: Falta de valor agregado	-	Automatización de extremo a extremo del flujo	Horas-persona al mes
P2: Errores humanos	L1	Validación de los datos extraídos y normalización de conceptos	Precisión de extracción por campo
P3: Variabilidad de formatos	L1, L2	Tolerancia a los cambios de formato	Precisión en formatos modificados o nuevos
P4: Fragilidad y costo de mantención	L1, L2, L7	Mantenibilidad y extensibilidad	Horas-persona de mantención y por formato nuevo
P5: Escalabilidad limitada	L2, L5	Escalabilidad ante el *peak* mensual	Tiempo de procesamiento del lote y costo por factura
P6: Latencia y baja visibilidad	L3, L4, L6	Rendimiento, aislamiento de fallas y observabilidad	Tiempo entre la llegada de la factura y su carga, estado consultable por factura

1.7 COMPETENCIAS EXISTENTES

Antes de plantear el desarrollo de una solución propia es necesario preguntarse por qué no se recurrió a una herramienta ya disponible en el mercado. La extracción automática de datos desde facturas es un problema conocido, para el cual existe una oferta amplia que puede agruparse en cinco categorías.

Tabla 6: Categorías de soluciones existentes para la extracción de datos desde facturas. Fuente: Elaboración propia.

Categoría	**Ejemplo**	**Fortaleza**	**Debilidad**
Extractores documentales en modalidad de servicio	Parseur, PDFelement	Puesta en marcha rápida y sin desarrollo	Operan documento a documento con un costo por uso, y no se integran con el sistema receptor (EMER) ni con la lógica de categorización de conceptos propia del negocio
Plataformas de automatización de bajo código	Microsoft Power Automate con AI Builder	Integración nativa con el repositorio documental (SharePoint) y la suite ofimática ya utilizada (Microsoft 365)	Su modelo de créditos añade un costo recurrente y la capacidad queda sujeta a los planes contratados por la organización
Servicios de extracción documental en la nube	Amazon Textract, Azure AI Document Intelligence, Google Document AI	Alta precisión de extracción y acceso mediante interfaces de programación	Resuelven únicamente la extracción, no cubren la captura, la validación, la categorización, la carga ni el seguimiento del estado, que son la mayor parte del proceso
Automatización robótica de procesos (*Robotic Process Automation, RPA*)	UiPath, Automation Anywhere	Permite automatizar la operación sobre la interfaz del sistema receptor sin modificarlo	Requiere licencias de costo elevado, resulta frágil ante cambios en la interfaz y no resuelve el problema de fondo, que es la extracción

Bibliotecas de código abierto basadas en reglas	Tesseract, pdfplumber, PyMuPDF	Sin costo de licencia y con control total sobre la lógica	Exigen un desarrollo por cada formato, es exactamente el enfoque de la automatización actual, cuyos límites ya fueron medidos (**L1, L2**)

Esta evaluación no es únicamente teórica. La empresa recibió en su momento la propuesta de un proveedor externo de software para hacerse cargo del proceso, que fue descartada por su costo y por el escaso control que ofrecía sobre el desarrollo, en un momento en que Evol buscaba precisamente consolidar un área de tecnología propia tras resultados insatisfactorios con la externalización. También se realizaron intentos concretos con una plataforma de automatización de bajo código (PowerAutomate con IA Builder), que se detuvieron al constatar que excedían los créditos y accesos disponibles en las cuentas corporativas y suponían un costo adicional que no se quiso asumir.

A ello se suma una restricción organizacional, al formar parte de un grupo mayor (LipiAndes), Evol se rige por estándares de seguridad corporativos que condicionan la modificación de permisos sobre las cuentas y los equipos, y que restringen el uso de herramientas de inteligencia artificial de terceros cuando no es posible verificar el control sobre la información procesada. Cualquier alternativa que dependa de enviar las facturas a un servicio externo no controlado por la empresa queda fuera de consideración.

En conjunto, ninguna de las categorías anteriores cubre por sí sola el proceso completo (desde el documento disponible en el repositorio hasta el dato registrado y su estado), ni permite conservar el control sobre la lógica de negocio que la categorización de conceptos exige. Lo que se requiere no es una herramienta puntual, sino una arquitectura que integre y coordine varios componentes, lo que justifica el enfoque adoptado en este trabajo.

1.8 OBJETIVOS

1.8.1 OBJETIVO GENERAL

Diseñar y validar una arquitectura de microservicios en la nube que automatice la extracción y carga de los datos de facturas en el sistema EMER de Evol Services, con el fin de reducir el esfuerzo manual y los errores del proceso actual.

1.8.2 OBJETIVOS ESPECÍFICOS

1. Analizar el proceso actual de carga de facturas y la automatización existente, midiendo su desempeño e identificando sus limitaciones, para definir los requerimientos que debe cumplir la nueva solución.
2. Seleccionar los servicios en la nube para cada componente de la solución mediante criterios explícitos, considerando la infraestructura tecnológica existente en Evol Services.
3. Diseñar la arquitectura de la solución, dividiendo el proceso en microservicios independientes y documentándola mediante diagramas y decisiones justificadas según los requerimientos definidos.
4. Implementar los servicios que componen la arquitectura, incluyendo el seguimiento y reporte del estado de procesamiento de cada factura.
5. Validar la solución comparándola con el proceso manual y con la automatización preliminar, mediante métricas de precisión de extracción, tiempo de procesamiento, esfuerzo de mantención y costo operacional.

1.9 ALCANCE

El presente trabajo aborda las etapas de revisión y carga del proceso de gestión de facturas, tomando como punto de entrada los documentos ya disponibles en el repositorio documental de la empresa (SharePoint) y como punto de salida su registro en el sistema EMER.

1.9.1 DENTRO DEL ALCANCE

- Las **facturas** de las distribuidoras y suministradoras **prioritarias**. La priorización sigue dos criterios sucesivos: en primer lugar, el número de facturas mensuales por emisor, de mayor a menor. Y en segundo lugar, ante volúmenes similares, la relevancia de los clientes asociados, determinada principalmente por su consumo de la instalación. Este orden maximiza primero el volumen cubierto y, a igualdad de condiciones, el impacto económico.
- Los documentos en **formato PDF digital con texto seleccionable**, que constituyen la totalidad de las facturas recibidas actualmente.
- El **seguimiento y reporte** del estado de procesamiento de cada factura, en su versión mínima, entendida como la capacidad de consultar qué ocurrió con cada documento.

- La **validación de la solución** en un ambiente de pruebas (*QA*) **con facturas reales**, comparándola con el procedimiento manual y con la automatización actual.

1.9.2 FUERA DE ALCANCE Y TRABAJO FUTURO

- La obtención de las facturas desde los portales de los emisores o desde el correo electrónico, que continúa realizándose como hasta ahora.
- Los documentos protegidos con contraseña, cuya apertura se resuelve antes de ingresar al flujo, y los documentos escaneados, prácticamente inexistentes en la operación actual.
- Las **facturas compuestas por más de un documento**, que deben cargarse de forma unificada para que el Reporte de Validación se elabore una sola vez sobre el total. Su incorporación no exige únicamente leer un formato adicional, sino esperar a que el conjunto esté completo antes de cargar, determinar cuándo lo está y qué hacer si uno de los documentos nunca llega, lo que constituye un problema de coordinación distinto del que aborda este trabajo. La solución reconoce estos casos y los deriva a revisión manual en lugar de cargarlos, de modo que no se registre un cobro incompleto. El método de agregación que permitiría unificarlos se describe en el Capítulo 3 y se declara como trabajo futuro.
- El caso del emisor cuya factura se acompaña de una proforma en formato de planilla (Excel) con el detalle de cargos reliquidados. Su incorporación no exige únicamente leer un formato distinto, sino resolver la correlación de dos documentos que llegan en momentos diferentes y determinar cuándo el conjunto está completo, lo que constituye un problema de coordinación propio y ajeno al objetivo de este trabajo.
- Las distribuidoras y suministradoras no prioritarias, que se incorporan mediante el mismo mecanismo diseñado y cuyo costo marginal de incorporación se mide en el Capítulo 4.
- Un panel interactivo de operación para los analistas, más allá de la consulta de estado señalada.
- La modificación de EMER para consumir directamente los datos, que se mantiene como sistema receptor sin alteraciones en su interfaz de carga.
- Otros tipos de documento distintos de las facturas eléctricas.

Respecto de la automatización actual, la solución propuesta la reemplaza de forma progresiva, durante el desarrollo de este trabajo ambas coexisten, atendiendo cada una los formatos que le corresponden, y la sustitución completa queda condicionada a los resultados de la validación.

1.9.3 ENTREGABLES

La arquitectura documentada mediante diagramas y decisiones justificadas, los servicios implementados para los formatos comprendidos en el alcance, el mecanismo de seguimiento del estado de cada factura y los resultados de la validación comparada.

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 ARQUITECTURA DE *SOFTWARE* Y ATRIBUTOS DE CALIDAD

Esta sección fija el vocabulario con el que se justifica el diseño del Capítulo 3 y se mide el resultado en el Capítulo 4. Es la base conceptual del aporte central de este trabajo.

2.1.1 QUÉ ES LA ARQUITECTURA DE *SOFTWARE*

La arquitectura de *software* de un sistema es el conjunto de estructuras necesarias para razonar sobre él, compuestas por elementos de *software*, las relaciones entre ellos y las propiedades de ambos (Bass, Kazman, & Clements, 2021). La definición es deliberadamente abstracta, la arquitectura no es un diagrama ni un conjunto de tecnologías, sino el conjunto de decisiones que resultan difíciles de revertir una vez tomadas, y que determinan si el sistema podrá o no cumplir las exigencias que se le imponen.

De ahí salen dos consecuencias pertinentes a este trabajo. La primera es la distinción entre arquitectura, que define los elementos y sus relaciones, diseño detallado, que resuelve el interior de cada elemento, e implementación, que lo construye, son decisiones de distinto alcance y distinto costo de corrección. La segunda es que un sistema puede funcionar y aun así tener una arquitectura deficiente. Es exactamente el caso descrito en el Capítulo 1, la automatización actual opera en producción y procesa cerca del 70% de las facturas, y sus problemas no están en que el código no funcione, sino en decisiones estructurales (lógica específica acoplada en un proceso único, alojado junto al sistema receptor) que hoy no se pueden revertir sin rediseñar.

2.1.2 ATRIBUTOS DE CALIDAD

Un atributo de calidad es una propiedad medible que indica qué tan bien un sistema satisface las necesidades de sus interesados, más allá de lo que hace (Bass, Kazman, & Clements, 2021). Los requerimientos funcionales describen las funciones del sistema y pueden satisfacerse con casi cualquier arquitectura, son los atributos de calidad los que discriminan entre alternativas arquitectónicas. Esto explica por qué el Capítulo 1 concluye que el problema no se resuelve programando más extractores, la función requerida ya se cumple, y lo que falla es la calidad con la que se hace.

La norma (ISO/IEC, 2023) ofrece un modelo de referencia que organiza la calidad de un producto de software en nueve características, adecuación funcional, eficiencia de desempeño, compatibilidad, capacidad de interacción, fiabilidad, seguridad de la información, mantenibilidad, flexibilidad e inocuidad, cada una con sus subcaracterísticas. La edición de 2023 incorporó la inocuidad y renombró dos características de la edición anterior, de modo que la escalabilidad pasó a ser una subcaracterística de la flexibilidad. No corresponde recitarla, lo pertinente es seleccionar explícitamente las que este trabajo compromete y declarar de dónde proviene cada una. La Tabla 7 recoge esa selección y su trazabilidad hacia los problemas y limitaciones del Capítulo 1.

Enunciar un atributo, sin embargo, no basta para diseñar ni para validar, decir que un sistema debe ser "mantenible" no permite comprobar si lo es. Por eso los atributos se expresan como escenarios de calidad, en el formato: estímulo, respuesta y medida (Bass, Kazman, & Clements, 2021) siento un estímulo que llega al sistema, la respuesta esperada ante él y una medida que permita verificarla. Un ejemplo del caso: un emisor modifica el diseño de su factura sin aviso previo (estímulo); el sistema extrae igualmente los campos requeridos o, si no lo consigue, deriva la factura a revisión indicando el motivo, sin cargar datos incorrectos (respuesta); la precisión por campo se mantiene sobre el umbral definido y ninguna factura se carga con un dato erróneo (medida). Escrito así, el atributo se vuelve verificable. El Capítulo 3 formula los escenarios y el Capítulo 4 los valida.

Cabe señalar que este planteamiento da continuidad a lo expuesto al cierre de la sección 1.3.3, donde se enunciaron cuatro exigencias para la solución: seguir funcionando cuando los formatos cambien, incorporar nuevos emisores con bajo esfuerzo, aislar las fallas de cada etapa y entregar visibilidad del estado de cada factura. Esas cuatro exigencias son atributos de calidad en el sentido aquí definido, y la Tabla 7 las traduce al vocabulario de la norma.

2.1.3 MÉTODOS DE DISEÑO Y DOCUMENTACIÓN

Tres instrumentos, de uso extendido en el ámbito profesional, estructuran el Capítulo 3.

El diseño dirigido por atributos (*Attribute-Driven Design, ADD*) es un método que deriva las decisiones de diseño a partir de los atributos de calidad requeridos, mediante iteraciones sucesivas de mejora en las que cada decisión responde a una exigencia declarada (Cervantes & Kazman, 2016). Se adopta en una versión ligera, interesa su lógica (decidir a partir de atributos, no de preferencias) y la trazabilidad que impone entre requerimiento y decisión.

El modelo C4 (Brown, n.d.) documenta una arquitectura en cuatro niveles de abstracción sucesivos: contexto, que sitúa el sistema frente a sus usuarios y sistemas vecinos; contenedores, que muestra las unidades desplegables; componentes, que abre el interior de cada una; y código, que rara vez se documenta. Este trabajo emplea los tres primeros. Su valor está en ofrecer un lenguaje estándar, con niveles de detalle explícitos, en lugar de un diagrama informal cuya interpretación dependa de quién lo lea.

Un registro de decisión arquitectónica (*Architecture Decision Record, ADR*) documenta de forma breve una decisión bajo el formato: contexto, alternativas evaluadas, decisión, consecuencias. Su utilidad no es administrativa, dado que la arquitectura se define por las decisiones difíciles de revertir, dejar constancia de por qué se tomó cada una, y de qué se aceptó a cambio, forma parte del entregable arquitectónico. El Capítulo 3 documenta así sus decisiones principales.

Tabla 7: Atributos de calidad seleccionados, según ISO/IEC 25010:2023 y Capítulo 1. Fuente: Elaboración propia a partir de ISO/IEC 25010:2023.

Carácterística	**Subcaracterísticas pertinentes**	**Por qué se compromete en este trabajo**	**Origen**
Adecuación funcional	Corrección funcional	Los datos cargados deben corresponder a los de la factura, son el insumo de la validación de cobros	P2, L1
Mantenibilidad	Modificabilidad, capacidad de prueba, analizabilidad	Incorporar o ajustar un formato sin desplegar el sistema completo, y poder determinar qué ocurrió con cada factura	P4, L6, L7
Flexibilidad	Escalabilidad, adaptabilidad	Absorber más emisores y el *peak* mensual sin que el esfuerzo crezca en proporción	P5, L2, L5
Fiabilidad	Tolerancia a fallos, recuperabilidad	Una falla en una etapa no debe detener el proceso ni perder documentos, y el reproceso no debe depender de una intervención manual	L3
Eficiencia de desempeño	Comportamiento temporal	El dato debe estar disponible sin esperar a la siguiente ejecución programada	P6, L4
Seguridad	Confidencialidad	Las facturas no deben salir del entorno controlado por la empresa	Restricción organizacional

2.2 ESTILOS ARQUITECTÓNICOS

Un estilo arquitectónico es un patrón de organización de alto nivel que define qué tipos de elementos componen un sistema y cómo se relacionan. Elegir uno no es una preferencia técnica, determina qué atributos de calidad el sistema podrá cumplir y cuáles le resultarán costosos. Esta sección presenta los tres estilos necesarios para leer el Capítulo 3: el monolítico, que describe la automatización actual; el de microservicios, que orienta la propuesta; y el orientado a eventos, que define cómo se comunican sus partes.

2.2.1 ARQUITECTURA MONOLÍTICA

Una arquitectura monolítica es aquella en que la aplicación se construye y se despliega como una sola unidad, con un único ciclo de vida, todos sus componentes comparten proceso, repositorio y despliegue (Richards & Ford, 2020).

Sus ventajas son reales y conviene reconocerlas, porque explican por qué la automatización actual fue una decisión razonable ante una urgencia operativa, es simple de construir y de desplegar, no hay latencia de red entre sus partes y permite depurar el flujo completo en un solo lugar. En martinfowler.com existe un artículo sobre microservicios (Lewis & Fowler, 2014) donde recomiendan de hecho comenzar por un monolito modular antes de distribuir un sistema, porque los límites entre servicios son difíciles de corregir una vez establecidos.

Sus limitaciones aparecen con el crecimiento: el acoplamiento hace que un cambio local propague efectos a distancia; cualquier modificación obliga a desplegar la totalidad; el escalado es indivisible, de modo que no se puede proporcionar más recursos sólo a la etapa que los necesita; y una falla del proceso o del servidor alcanza a todo el sistema a la vez.

La correspondencia con el Capítulo 1 es directa. La limitación L7 describe un proyecto en que el 73% del código es lógica específica por formato, sin separación de responsabilidades, modificar un solo extractor exige desplegar el conjunto. La limitación L3 muestra el otro efecto, al estar el proceso en el mismo servidor que el sistema receptor, la saturación de ese servidor deja el lote completo sin procesar. Es importante precisar este punto, una factura que falla no detiene a las siguientes, de modo que el problema no es la propagación del error entre documentos, sino la ausencia de aislamiento entre etapas y entre sistemas.

2.2.2 ARQUITECTURA DE MICROSERVICIOS

Una arquitectura de microservicios organiza el sistema como un conjunto de servicios pequeños, cada uno con una responsabilidad acotada, desplegables de forma independiente, que administran sus propios datos y se comunican mediante interfaces explícitas (Lewis & Fowler, 2014) (Newman, 2021). De las características que Lewis y Fowler enumeran, cuatro resultan adecuadas a este trabajo: la componentización mediante servicios y no mediante bibliotecas, el despliegue independiente, el diseño para la falla (el sistema debe seguir operando cuando un servicio deja de responder) y el diseño evolutivo, que asume que cada servicio será reemplazado antes que el sistema completo.

Los beneficios responden a las limitaciones del apartado anterior, pero tienen un costo que debe declararse, y así saber si el estilo se justifica para este volumen. La Tabla 8 los contrasta. La distribución introduce complejidad operacional, convierte una llamada en memoria en una llamada de red con su propia latencia y sus modos de falla, deja la consistencia de los datos sin el respaldo de una transacción única, y exige una infraestructura de monitoreo que un monolito no necesita para depurar un flujo que atraviesa varios servicios. Lewis y Fowler advierten además que el resultado depende

fuertemente de la capacidad del equipo, y que uno sin la experiencia necesaria puede producir un sistema distribuido peor que el monolito que reemplaza.

Respecto de la descomposición, los criterios habituales son dividir por capacidad de negocio o por etapa del flujo de procesamiento (Newman, 2021). El Capítulo 3 utiliza el segundo y justifica los límites elegidos.

2.2.3 ARQUITECTURA ORIENTADA A EVENTOS

En una arquitectura orientada a eventos los componentes se comunican mediante eventos, es decir, notificaciones de que un hecho relevante ocurrió. El emisor genera el evento sin conocer quién lo consumirá ni cuándo, y el consumidor lo procesa cuando puede. De ahí resulta un desacoplamiento doble, temporal y espacial, las partes no necesitan estar disponibles al mismo tiempo ni conocerse entre sí (Richards & Ford, 2020).

Las colas de mensajes materializan ese desacople y cumplen tres funciones: absorben los *peaks* de carga, acumulando trabajo que se procesa al ritmo que el sistema permite; permiten el procesamiento asíncrono y paralelo de elementos independientes; y regulan el caudal hacia un consumidor lento, protegiéndolo de recibir más solicitudes de las que puede atender. Esta última función importa cuando el destino de los datos es un sistema preexistente que no se puede modificar.

Las colas también imponen una exigencia. Los servicios de mensajería habituales garantizan la entrega al menos una vez, lo que significa que un mismo mensaje puede entregarse más de una vez y que el orden no está asegurado (Amazon Web Services, 2026a). En consecuencia, el consumidor debe ser idempotente: una operación es idempotente cuando ejecutarla varias veces produce el mismo efecto que ejecutarla una sola. Aplicado a este caso, reprocesar una factura no debe registrar dos veces sus cargos ni gatillar dos veces la cadena posterior de reportes descrita en el Capítulo 1. La idempotencia no es un refinamiento opcional, es la condición que permite reintentar con seguridad.

Sobre esta se apoya el manejo de errores. Un reintento vuelve a ejecutar una operación que falló, habitualmente con esperas crecientes entre intentos. Una cola de mensajes fallidos (*dead-letter queue , DLQ*) toma los mensajes que llegaron al límite de reintentos, para su inspección y reproceso posterior sin pérdida de información. La diferencia que determina el diseño es la que separa los errores transitorios que el reintento resuelve (una indisponibilidad momentánea, una restricción de cantidad de recepciones, un tiempo de espera agotado), de los permanentes (un documento ilegible, una respuesta que no cumple el esquema esperado), en los que reintentar solo repite el gasto y el fallo.

Por último, la coordinación entre servicios admite dos formas. En la coreografía, cada servicio reacciona a los eventos que le conciernen y ningún componente conoce el flujo completo, en la orquestación, un componente explícito dirige la secuencia y conserva el estado de cada ejecución (Newman, 2021). La coreografía maximiza el desacoplamiento, la orquestación facilita saber en qué etapa está cada elemento y reintentar una etapa

concreta, lo que la hace pertinente cuando la visibilidad del proceso es un requisito y no una comodidad. La coreografía también puede ofrecer esa visibilidad, pero solo si cada servicio registra su propio avance. El Capítulo 3 explica cuál se adopta y por qué.

La relación con el Capítulo 1 cierra esta sección. La limitación L4 describe una latencia de hasta veinticuatro horas originada en una ejecución programada una vez al día, y el problema P6 suma la falta de visibilidad del estado de cada factura. Un flujo gestionado por eventos reemplaza la espera a la próxima ejecución por el procesamiento en cuanto el documento está disponible, y un flujo orquestado registra por dónde va cada factura, son atributos que el estilo aporta, no funcionalidades que haya que programar.

Tabla 8: Beneficios y costos de una arquitectura de microservicios. Fuente: Elaboración propia a partir de Lewis y Fowler (2014), Newman (2021) y Richards y Ford (2020).

Dimensión	**Beneficio**	**Costo asociado**
Despliegue	Cada servicio se despliega sin afectar a los demás	Más componentes que construir, versionar y coordinar
Escalado	Se asignan recursos solo a la etapa que los requiere	Requiere definir y ajustar límites por servicio
Fallas	Una falla queda contenida en su servicio	Aparecen fallas nuevas, propias de la red
Comunicación	Interfaces explícitas entre partes	Latencia de red y serialización en cada salto
Datos	Cada servicio administra su propia persistencia	La consistencia deja de estar garantizada por una transacción
Evolución	Un servicio se reemplaza sin reescribir el sistema	Los límites entre servicios son difíciles de corregir
Operación	Responsabilidades acotadas y comprensibles	Exige observabilidad y automatización del despliegue
Equipo	Permite trabajar en paralelo sobre partes distintas	El resultado depende de la experiencia del equipo

2.3 COMPUTACIÓN EN LA NUBE

La sección anterior se describió cómo se organiza un sistema, esta describe el terreno sobre el que se construye. Su propósito es definir los modelos de servicio que explican la estructura de costos de la situación actual, caracterizar el modelo de ejecución que adopta la propuesta junto con sus límites, y presentar los servicios que el Capítulo 3 selecciona, cada uno con el rol que cumple.

2.3.1 MODELO DE SERVICIOS

El documento de referencia en la materia define la computación en la nube como un modelo que permite el acceso por red, conveniente y bajo demanda, a un conjunto compartido de recursos de cómputo configurables, que pueden aprovisionarse y liberarse con un esfuerzo mínimo de gestión (Mell & Grance, 2001). Ese modelo se reconoce por cinco características esenciales: autoservicio bajo demanda, acceso amplio a la red, agrupación de recursos, elasticidad rápida y servicio medido, esta última entendida como la capacidad de cuantificar el consumo y facturar en función de él.

El mismo documento distingue tres modelos de servicio según qué administra el proveedor y qué queda a cargo del cliente. En infraestructura como servicio (*Infrastructure as a Service, IaaS*) el proveedor entrega máquinas virtuales, almacenamiento y red, y el cliente administra el sistema operativo y todo lo que corre sobre él. En plataforma como servicio (*Platform as a Service, PaaS*) el proveedor administra además la plataforma de ejecución. En software como servicio (*Software as a Service, SaaS*) entrega una aplicación terminada. Luego de esa clasificación se consolidó un cuarto modelo, funciones como servicio (*Funtions as a Service, FaaS*), en que la unidad de despliegue es una función que el proveedor ejecuta en respuesta a un evento, sin que el cliente administre servidor alguno.

Situar la automatización actual en esta clasificación explica una de sus limitaciones. El bot se ejecuta sobre *IaaS*: una instancia virtual permanentemente encendida, administrada por la empresa. De ahí se sigue el costo fijo que L5 describe, independiente de que la carga se concentre en unos pocos días del mes, junto con la responsabilidad de mantener el sistema operativo y de recuperar el servicio cuando la instancia falla.

2.3.2 EL MODELO *SERVERLESS*

El término *serverless* designa un modelo de ejecución en que el proveedor administra por completo la infraestructura, no hay servidores que aprovisionar ni escalar, la plataforma crea tantas instancias de ejecución como invocaciones concurrentes reciba, y la facturación se calcula por invocación y por tiempo de ejecución consumido, sin costo cuando no hay actividad. El nombre es engañoso (los servidores existen, pero son invisibles para quien despliega) y su consecuencia relevante es el traslado del costo fijo de infraestructura a un costo variable por uso.

El modelo tiene límites que deben declararse, porque condicionan lo que se puede diseñar sobre él:

- **Arranque en frío.** Cuando no hay una instancia de ejecución disponible, la plataforma debe crear una antes de atender la invocación, lo que añade latencia. Golec, et al. (2024) sistematizan el fenómeno y lo identifican como una consecuencia directa de la capacidad de escalar a cero, es decir, del mismo mecanismo que permite no pagar por inactividad.
- **Tiempo máximo de ejecución.** Una invocación no puede extenderse indefinidamente; en la plataforma utilizada el máximo es de 900 segundos, 15 minutos (Amazon Web Services, 2026b).
- **Memoria y almacenamiento temporal acotados.** La memoria asignable va de 128 a 10.240 MB, y el almacenamiento temporal en disco es del mismo orden.
- **Tamaño de la carga útil (*payload*).** Lo que entra y sale de una invocación tiene un tope: 6 MB en invocaciones sincrónicas y 1 MB en asincrónicas. Ese límite es pertinente cuando lo que se procesa son documentos, y obliga a pasar los archivos por referencia en lugar de incluirlos en el mensaje.
- **Dependencia del proveedor.** Un flujo construido sobre servicios administrados queda ligado a las interfaces de ese proveedor. La consecuencia debe asumirse de forma explícita, el Capítulo 3 la documenta como una decisión con consecuencias registradas y no como un efecto colateral.

A cambio de esos límites, el pago por uso resulta con ventaja en un perfil de carga preciso, el de las cargas intermitentes y concentradas. Cuando la actividad se agrupa en pocos días y el resto del mes es de inactividad, una infraestructura dimensionada para el máximo permanece en desuso la mayor parte del tiempo. Ese es exactamente el perfil descrito en el Capítulo 1, con la recepción de facturas concentrada entre la segunda y la tercera semana de cada mes.

2.3.3 SERVICIOS UTILIZADOS

Esta parte presenta las categorías de servicios que compone la arquitectura del Capítulo 3. Cada una se enuncia con la fórmula qué es y qué rol cumple, para evitar que la sección se convierta en un catálogo. La Tabla 9 las reúne.

Antes de continuar, es necesario aclarar el contexto. La elección del proveedor no forma parte de este trabajo, es una restricción. Evol Services opera sobre *Amazon Web Services* por decisiones previas (un contrato vigente, la experiencia del equipo de Desarrollo y Tecnología, y el hecho de que el sistema receptor de los datos ya está alojado allí), a lo que se suma la restricción de seguridad corporativa descrita en la sección 1.7, que excluye el envío de documentos a servicios de terceros no gobernados por la empresa. El mérito del segundo objetivo específico no está en comparar proveedores de nube, sino en seleccionar los servicios adecuados dentro del proveedor disponible. Cabe señalar que las categorías de la Tabla 9 son genéricas y tienen equivalentes en las demás nubes de uso extendido, de modo que el diseño resultante es conceptualmente portable aunque su implementación no lo sea.

Tabla 9: Categorías de servicios utilizadas y el rol que cumplen en este trabajo. Fuente: Elaboración propia.

Categoría de servicio	**Qué es**	**Qué rol cumple**
Almacenamiento de objetos	Repositorio de archivos accesibles por clave, con costo por volumen almacenado y reglas de ciclo de vida	Recibe el documento que ingresa al flujo y conserva los archivos de cada etapa: el archivo original, el texto extraído y el resultado de la extracción
Cómputo por funciones (*FaaS*)	Ejecución de código en respuesta a un evento, sin administrar servidores	Aloja cada uno de los servicios de la arquitectura, con despliegue y escalado independientes
Cola de mensajes	Almacenamiento temporal de mensajes entre un emisor y un consumidor, con reintentos y cola de mensajes fallidos	Comunica las etapas, amortigua el *peak* mensual de recepción y regula el envío hacia el sistema receptor
Reconocimiento documental	Servicio administrado que extrae texto, tablas y estructura desde un documento	Convierte el PDF de la factura en texto, y deja el flujo preparado para documentos escaneados
Modelos de lenguaje administrados	Acceso a modelos de lenguaje (*LLM*) dentro de la propia cuenta del cliente, sin infraestructura propia	Interpreta el texto de la factura y devuelve los datos en una estructura definida.
Base de datos NoSQL documental	Almacenamiento de documentos de estructura variable, con acceso directo por clave	Guarda el estado de procesamiento de cada factura y los metadatos del proceso
Gestión de identidades y permisos	Definición de qué puede hacer cada componente sobre cada recurso	Otorga a cada servicio únicamente los permisos que su función requiere
Observabilidad	Recolección de registros, métricas y alarmas sobre la operación	Provee la visibilidad cuya ausencia constituye la limitación L6

Infraestructura como código (*Infrastructure as Code, Iac*)	Declaración de los recursos de infraestructura en archivos versionados, desplegables de forma reproducible	Permite versionar la arquitectura junto con el código y reproducir el entorno, lo que contribuye a la mantenibilidad

2.3.4 PERSISTENCIA: RELACIONAL FRENTE A NOSQL

Una base de datos relacional organiza la información en tablas de esquema fijo, definido antes de insertar datos y validado en cada escritura, con relaciones explícitas entre tablas y un lenguaje de consulta que permite combinarlas. Es el modelo adecuado cuando la estructura de los datos es estable y conocida, y cuando las consultas cruzan varias entidades. Una base de datos NoSQL documental (clave-valor) almacena documentos autocontenidos, habitualmente en formato JSON, cuya estructura puede variar de un documento a otro sin necesidad de declararla de antemano, privilegia el acceso directo por clave y la escritura de estructuras anidadas sobre la capacidad de combinar entidades.

Para la persistencia intermedia de datos extraídos de facturas, el modelo documental presenta ventajas concretas que se siguen de lo expuesto en 2.4.1: el número de cargos varía entre facturas, hay campos que existen en unos emisores y no en otros, y la estructura de lo extraído cambia cada vez que se incorpora un formato nuevo. Un esquema fijo obligaría a modificar la base ante cada una de esas variaciones, en cambio un esquema flexible las absorbe. A esto se agrega que el registro del estado de procesamiento de cada factura (la información que el cuarto objetivo específico exige poder consultar y cuya ausencia se lleva a la limitación L6) corresponde a un acceso directo por identificador, que es precisamente el patrón para el que este tipo de base está optimizado.

Hay un límite que condiciona el diseño: las bases documentales imponen un tamaño máximo por documento, de modo que los elementos de mayor volumen (el archivo original, el texto extraído completo) no deben almacenarse en ellas. La práctica establecida consiste en guardar en la base los metadatos y el estado, junto con referencias a los elementos, y estos en un servicio de almacenamiento de objetos, de costo por GB mucho menor. El Capítulo 3 detalla ese reparto.

Finalmente, conviene hacer una precisión para evitar una interpretación equivocada. El sistema EMER, destino final de los datos, es y seguirá siendo relacional, y este trabajo no propone cambiarlo, allí los datos ya están normalizados, sus relaciones son estables y se consultan de forma cruzada para elaborar los reportes. Lo que aquí se discute es la persistencia intermedia del proceso de extracción, que tiene requisitos distintos (estructura variable, esquema en evolución, acceso por identificador) y que por eso admite una decisión distinta. Los dos modelos coexisten porque resuelven problemas diferentes.

2.4 EXTRACCIÓN DE DATOS DESDE DOCUMENTOS

El Capítulo 1 dejó planteado un hallazgo que esta sección debe explicar, la automatización actual cuenta con un extractor para cada uno de los treinta formatos vigentes y, aun así, aproximadamente una de cada tres facturas no logra procesarse. Ese resultado no viene de una implementación deficiente, sino de una propiedad del enfoque empleado.

2.4.1 EL DOCUMENTO SEMIESTRUCTURADO

Un dato estructurado reside en un esquema explícito, definido antes que el dato mismo, como ocurre en una tabla relacional. Un dato no estructurado es falto de organización interna, como el texto libre de un correo. En un punto intermedio de ambos se ubican los datos semiestructurados: la información presenta una organización identificable (campos etiquetados, tablas, agrupaciones), pero esa organización no está declarada en un esquema previo, no es uniforme entre documentos y puede variar en el tiempo (Abiteboul, 1997).

La factura es un caso característico. Todas contienen los mismos elementos (emisor, receptor, periodo facturado, detalle de cargos, totales), pero cada emisor decide cómo etiquetarlos, en qué orden presentarlos y con qué disposición, porque el documento está diseñado para que lo lea una persona y no para ser procesado por un *software*. El contenido es conceptualmente equivalente entre emisores, mas su representación no.

Conviene distinguir dos niveles de variabilidad, porque solo el segundo explica el comportamiento observado en el Capítulo 1. La variabilidad entre formatos es evidente, treinta emisores producen treinta disposiciones distintas. Pero la variabilidad dentro de un mismo formato es menos visible y más costosa: un mismo emisor, entre un periodo y otro, agrega o suprime cargos según el cliente, extiende una tabla, incorpora una reliquidación o ajusta su diseño. La primera se resuelve con cobertura, la segunda no. Por eso tener un extractor por formato no garantiza el procesamiento.

2.4.2 ENFOQUES TRADICIONALES: RECONOCIMIENTO ÓPTICO, REGLAS Y PLANTILLAS

El reconocimiento óptico de caracteres (*Optical Character Recognition, OCR*) convierte en texto los caracteres presentes en una imagen. Los vuelve legibles, pero no los interpreta, entrega una secuencia de caracteres y, según la herramienta, su posición en la página, pero no indica qué significa cada valor. Como las facturas del caso son documentos PDF digitales con texto seleccionable, el reconocimiento óptico no constituye el cuello de botella, aporta en cambio, un rol en el diseño, porque permite que el flujo admita documentos escaneados o fotografiados sin alterar su estructura.

La extracción basada en reglas y plantillas opera sobre el texto ya disponible y define, para cada formato, dónde buscar cada dato: coordenadas dentro de la página, expresiones regulares, referencias textuales o el recorrido de la estructura interna del documento. Es el enfoque de la automatización actual, y su virtud principal es el determinismo, ante la misma entrada produce siempre la misma salida, de modo que un error es reproducible y depurable.

Sus limitaciones están documentadas. Yashwant et al. (2025) caracterizan estos sistemas como frágiles y sin capacidad de generalizar a formatos nuevos o no vistos, y señalan que la carga de mantención es significativa porque cada disposición nueva exige una plantilla nueva. La coincidencia con lo observado en Evol Services es directa, L1 y L2 (Tabla 1) no describen una particularidad de la implementación de la empresa, sino una propiedad conocida del enfoque. De ahí siguen dos de los problemas del Capítulo 1: el esfuerzo crece de forma proporcional al número de emisores (P5), y cada regla añadida para cubrir un caso borde aumenta la complejidad ante futuros cambios (P4).

Los modelos de comprensión de documentos entrenados de forma supervisada son una alternativa intermedia entre los enfoques basados en reglas y los modelos generativos. Estos modelos aprenden a reconocer campos combinando el contenido textual con su posición en la página, como ocurre con LayoutLM (Xu, et al., 2019). Si bien reducen la dependencia de reglas explícitas, requieren un conjunto de documentos etiquetados campo por campo, lo que implica un esfuerzo de clasificación difícil de justificar cuando se tienen treinta formatos con volúmenes de documentos muy variados.

2.4.3 PROCESAMIENTO INTELIGENTE DE DOCUMENTOS Y MODELOS DE LENGUAJE DE GRAN ESCALA

Con el nombre de procesamiento inteligente de documentos (*Intelligent Document Processing, IDP*) se habla de la categoría de soluciones que conectan reconocimiento del contenido, interpretación y validación para convertir documentos en datos consumibles por otros sistemas. No es una técnica única, sino la integración de varias etapas en un flujo.

El aporte de los modelos de lenguaje de gran escala (*Large Language Model, LLM*) dentro de esa categoría consiste en un cambio en el criterio de extracción, de usar la posición a usar el significado. El modelo recibe el texto del documento junto con una instrucción que describe qué datos se requieren, e identifica cada valor por su sentido y no por la coordenada en que aparece. Un cargo por energía se reconoce por lo que representa, aunque el emisor lo haya renombrado o reubicado. Esto ataca la causa de L1 y reduce el costo de L2, porque incorporar un emisor deja de exigir la programación de un extractor. La misma instrucción puede declarar además la forma de la respuesta (un objeto JSON con campos y tipos definidos), lo que permite verificar la salida contra ese esquema antes de utilizarla. (Wang, et al., 2025) sitúan esta transición, desde los modelos especializados hacia los modelos generativos instruidos, como una de las principales tendencias en la evolución reciente del área.

El enfoque introduce riesgos propios, que condicionan el diseño y deben declararse:

1. **Alucinación.** El modelo puede devolver valores razonables, pero ausentes del documento. Es el mismo riesgo sobre el dato que el Capítulo 1 atribuye a la transcripción manual (P2), con otra causa y sin el criterio de un validador humano que lo detecte.
2. **Ausencia de referencia espacial.** Un *LLM* no indica de forma nativa la disposición espacial de la página ni conecta cada valor extraído con el lugar del documento del que proviene, lo que dificulta verificar su origen (Perot, et al., 2024).
3. **No determinismo.** Dos ejecuciones sobre la misma entrada pueden producir salidas distintas. El comportamiento se disminuye con una configuración adecuada, pero no se elimina, y con él se pierde la consistencia del enfoque por reglas.
4. **Costo y latencia por documento.** Ambos son proporcionales al texto procesado, de modo que el costo deja de ser fijo y pasa a depender del volumen.
5. **Límite de contexto.** El texto admitido en una sola invocación es finito, lo que impone un tratamiento particular a los documentos extensos.

La conclusión que estos riesgos imponen es la más relevante de la sección, la validación posterior a la extracción no es opcional. Un enfoque semántico sin validación cambia un error conocido y reproducible por uno impredecible, y lleva al cliente final las consecuencias de un dato incorrecto, según la secuencia descrita en el Capítulo 1. La literatura sobre extracción en el ámbito financiero coincide en considerar el control de las alucinaciones como parte del diseño del sistema, y no como un ajuste posterior (Sarmah et al., 2023). Esa exigencia justifica que la arquitectura del Capítulo 3 incorpore un componente de validación con responsabilidad propia.

Conviene distinguir, por último, los servicios administrados de extracción documental frente a los modelos de lenguaje (*LLM*) de propósito general. Los primeros devuelven texto, tablas y campos con una medida de confianza asociada, con comportamiento estable y costo predecible, pero se ajustan a los tipos de documento ya previstos por el proveedor; los segundos interpretan cualquier documento a partir de una instrucción (*prompt*), a cambio de los riesgos enumerados. No son excluyentes, combinar un servicio que entrega el texto con un modelo que lo interpreta es un patrón establecido, recogido en las arquitecturas de referencia que los proveedores de nube publican para el procesamiento inteligente de documentos y que se revisan en la sección 2.5.

Tabla 10: Comparación de enfoques de extracción de datos desde facturas. Fuente: Elaboración propia a partir de Yashwant et al. (2025), Xu et al. (2020) y Perot et al. (2023).

Criterio	**Reglas y plantillas**	**Modelos de comprensión de documentos**	**Modelos de lenguaje con instrucciones**
Criterio de extracción	Posición y patrones de texto	Texto y posición aprendidos de ejemplos	Significado del contenido
Costo de un emisor nuevo	Programar y validar un extractor	Etiquetar documentos y reentrenar	Ajustar la instrucción, si es necesario
Tolerancia al cambio de formato	Casi nula, el cambio rompe la regla	Parcial, según la cobertura del entrenamiento	Alta, no depende de la disposición espacial
Determinismo	Total	Alto	Limitado
Estructura del costo	Fijo, en horas de desarrollo	Fijo en clasificación, variable en cómputo	Variable, por documento procesado
Riesgo principal	Fragilidad ante la variabilidad	Esfuerzo de clasificación por formato	Alucinación y ausencia de referencias
Validación posterior	Recomendable	Recomendable	Indispensable

2.5 TRABAJOS RELACIONADOS

La automatización de la extracción de datos desde facturas no es un problema nuevo, de modo que corresponde situar este trabajo respecto de lo ya publicado y precisar qué lo distingue.

En el ámbito académico, Yashwant et al. (2025) abordan la extracción de campos desde facturas con herramientas basadas en modelos de lenguaje y proponen un marco de evaluación con métricas de precisión por campo, verificaciones de consistencia y coincidencia exacta. Su aporte principal para este trabajo es doble, el diagnóstico sobre la fragilidad de los enfoques por plantillas, ya citado en 2.4.2, y la idea de que la evaluación debe hacerse campo a campo y no como una tasa global de acierto. Khanchandani et al. (2026) presentan una plataforma que combina reconocimiento óptico (*OCR*), modelos de aprendizaje profundo (*Deep Learning, DL*) y modelos de lenguaje (*LLM*) para facturas de

disposición variable, con foco en documentos de baja calidad, no reportan, sin embargo, cifras comparativas contra una línea base. En una vista más amplia, el estudio de Wang et al. (2025) documenta la transición del área desde los modelos especializados hacia los modelos generativos instruidos, pero no aborda casos de integración con sistemas existentes.

En el ámbito de las arquitecturas de referencia, (Amazon Web Services, 2026c) publica una guía para procesamiento inteligente de documentos cuya estructura coincide en lo esencial con el enfoque de este trabajo, captura en almacenamiento de objetos, encolado con control de concurrencia, funciones para cada etapa y una base documental para el seguimiento del estado. Difiere en la coordinación: la guía dirige el flujo con un orquestador explícito, mientras que este trabajo encadena las etapas mediante colas, por las razones que expone el Capítulo 3. Su existencia respalda las decisiones del Capítulo 3, pero *AWS* muestra una plantilla genérica, no ligada a un proceso de negocio ni evaluada contra una operación real.

Tabla 11: Trabajos relacionados y su relación con esta memoria. Fuente: Elaboración propia.

Trabajo	**Enfoque de extracción**	**Estilo arquitectónico**	**Validación reportada**	**Diferencia con este trabajo**
Yashwant et al. (2025)	Herramientas basadas en modelos de lenguaje	No abordado	Métricas de precisión por campo sobre un conjunto de documentos	No diseña una arquitectura ni integra con un sistema receptor
Khanchandani et al. (2025)	Reconocimiento óptico, aprendizaje profundo y modelos de lenguaje	No abordado	Sin cifras comparativas	No declara línea base ni costo operacional
Wang et al. (2025)	Revisión del área	No abordado	No aplica	Panorama del campo, sin caso de aplicación
Amazon Web Services (2026c)	Reconocimiento documental y modelos de lenguaje	Serverless, orientado a eventos y orquestado	No aplica, es arquitectura de referencia	Plantilla genérica, sin proceso de negocio ni evaluación sobre una operación real

Esta memoria	Reconocimiento documental y modelo de lenguaje, con validación determinística posterior	Microservicios serverless, orientados a eventos	Comparación en ambiente de calidad contra dos líneas base en producción	-

Lo que este trabajo comparte con los anteriores es el enfoque técnico de extracción. Lo que lo distingue son cinco condiciones de su contexto, que ninguno de ellos enfrenta de forma conjunta: existe una línea base cuantificada y en producción contra la cual comparar, y no una comparación contra un proceso manual hipotético; el destino de los datos es un sistema interno heredado, con su propia interfaz de carga, que no se modifica; el origen es un repositorio documental corporativo en uso, con sus restricciones de acceso; hay restricciones organizacionales de seguridad que excluyen los servicios de terceros no controlados por la empresa; y el volumen es medio y muy concentrado en el tiempo, un perfil distinto del alto volumen continuo que suponen la mayoría de los casos publicados.

El aporte de esta memoria no consiste, por tanto, en superar a los trabajos revisados, sino en algo que ninguno de ellos ofrece, una arquitectura justificada mediante decisiones documentadas y validada contra una operación real, para un contexto con esas restricciones. Los capítulos siguientes desarrollan ese diseño y su validación.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

3.1 ANTECEDENTE Y REQUERIMIENTOS

El diseño que se presenta en este capítulo sigue la lógica del diseño dirigido por atributos descrita en la sección 2.1.3, en su versión ligera: primero se fijan los requerimientos y los atributos de calidad que la solución debe cumplir, luego se toman las decisiones arquitectónicas que responden a ellos y, por último, se detalla cada componente. Esta sección cumple el primer paso. Su punto de partida son los seis problemas del proceso (P1 a P6) y las siete limitaciones de la automatización actual (L1 a L7) identificados en el Capítulo 1, y su resultado es el conjunto de exigencias verificables contra el cual el Capítulo 4 valida la solución.

3.1.1 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales describen qué debe hacer la solución, con independencia de cómo se implemente. Se identifican once, que recorren el flujo de una factura desde su detección hasta su registro (Tabla 12). Tres de ellos merecen una precisión. RF5 establece que una factura dudosa nunca se descarta ni se carga a medias, se deriva a revisión con el motivo, lo que incluye a los emisores cuyas facturas se componen de más de un documento, excluidos del alcance en la sección 1.9. RF10 elimina la necesidad de volver a depositar un archivo para reintentarlo, que hoy es la única vía de reproceso. Y RF11, el traslado del documento a su carpeta definitiva, replica un comportamiento de la automatización actual y queda sujeto a los permisos que se obtengan sobre el repositorio documental. Conviene advertir, por último, que estos requerimientos atienden el problema P1 solo de forma parcial, la solución automatiza la revisión y la carga, pero el depósito inicial de cada factura en la carpeta de su formato forma parte de la obtención, que quedó fuera del alcance, y sigue siendo una tarea del analista.

Tabla 12: Requerimientos funcionales. Fuente: Elaboración propia.

ID	Requerimiento	Origen
RF1	Detectar, en un intervalo de minutos, las facturas nuevas depositadas en las carpetas de entrada	L4, P6
RF2	Copiar cada documento a un almacenamiento propio, identificar su emisor y registrarlo	L6
RF3	Extraer los datos de la factura en la estructura que EMER recibe	P3, L1
RF4	Validar los datos extraídos: estructura, coherencia aritmética, conceptos del emisor y duplicados	P2

RF5	Derivar a revisión, indicando el motivo, toda factura que no supere la validación o que corresponda a un emisor de documentos compuestos	P2, alcance 1.9
RF6	Cargar los datos en EMER mediante su interfaz existente, sin registrar dos veces la misma factura	P1
RF7	Registrar el estado y la etapa de cada factura, con sus marcas de tiempo	L6, P6
RF8	Permitir consultar el estado por factura y por filtros (emisor, periodo, estado)	L6
RF9	Enviar diariamente al equipo GCP un resumen del procesamiento	L6, P6
RF10	Reprocesar una factura sin volver a depositarla	L6
RF11	Trasladar el documento cargado a su carpeta definitiva en el repositorio documental	Continuidad Operativa

3.1.2 ESCENARIOS DE CALIDAD

Los atributos seleccionados en la Tabla 7 se expresan ahora como escenarios de calidad en el formato estímulo, respuesta, medida (sección 2.1.2), a los que se agrega una meta. La meta convierte la validación del Capítulo 4 en una verificación, no basta con mostrar que la solución mejora respecto de las líneas base, sino que debe comprobarse si alcanza lo comprometido. La Tabla 13 reúne los ocho escenarios.

Las metas se derivan de la línea base de la Tabla 4 y de la naturaleza de cada atributo. La de mantenibilidad (E2) exige que incorporar un formato cueste a lo más un cuarto de las cerca de 25 horas que hoy demanda un extractor nuevo, y que el cambio quede en un único servicio. La de desempeño (E5) sustituye la espera de hasta 24 horas por un máximo de una hora entre el depósito de la factura y la disponibilidad del dato en EMER, en condiciones normales de operación. Para el *peak* mensual la meta es operativa: las facturas de un día de alta recepción (del orden de cien, según el equipo GCP) deben quedar cargadas dentro de la misma jornada, es decir, en a lo más ocho horas desde su depósito. Las metas de fiabilidad (E3 y E4) y de visibilidad (E6) son absolutas, ninguna factura perdida, ninguna afectada por una falla ajena, todas con su estado consultable, porque describen justamente las propiedades cuya ausencia documentó el Capítulo 1. La de adecuación funcional (E1) combina una meta relativa, no ser menos precisa que la automatización actual, con una absoluta: ningún dato llega a EMER sin haber superado la validación, lo que responde al riesgo de alucinación descrito en la sección 2.4.3.

Tabla 13: Escenarios de calidad de la solución. Fuente: Elaboración propia a partir de la Tabla 7 y de la línea base de la Tabla 4.

ID	Atributo	Estímulo	Respuesta esperada	Medida	Meta
E1	Adecuación funcional	Un emisor modifica el diseño de su factura sin aviso	Los datos se extraen igual o la factura se deriva a revisión con el motivo	Precisión por campo; facturas cargadas con un dato no validado	Precisión mayor o igual que la del bot; ninguna carga con un dato que no superó la validación
E2	Mantenibilidad (modificabilidad)	Se incorpora un formato nuevo	Se ajusta la configuración o la instrucción de un solo servicio, sin desplegar los demás	Horas de trabajo; servicios modificados	A lo más 6 horas (25% de las ~25 h del bot) y un solo servicio modificado
E3	Fiabilidad (tolerancia a fallos)	EMER no responde durante una carga	La carga se reintenta con espera; si los reintentos se agotan, la factura queda en error con su motivo	Facturas perdidas; facturas sin estado	Ninguna factura perdida; todas con estado registrado
E4	Fiabilidad (aislamien-to)	Una etapa falla al procesar una factura	Las demás facturas continúan su procesamiento	Facturas afectadas por una falla ajena	Ninguna
E5	Eficiencia de desempeño	Se deposita una factura fuera del peak mensual	Se procesa sin esperar una ejecución programada	Tiempo entre el depósito y el dato disponible en EMER	A lo más 1 hora. En el *peak*: un día de alta recepción (~100 facturas) cargado en a lo más 8 horas

E6	Mantenibilidad (analizabilidad)	El equipo GCP necesita saber qué ocurrió con una factura	El estado, la etapa y el motivo se consultan sin acudir al área de TI	Porcentaje de facturas con estado consultable	100%
E7	Seguridad (confidencialidad)	Se procesa una factura	El documento no sale del entorno gobernado por la empresa	Servicios externos que reciben el documento	Ninguno fuera del repositorio documental y de la cuenta de nube de la empresa
E8	Flexibilidad (escalabilidad)	Llega el peak mensual de facturas	Se absorbe sin intervención manual y con un costo que crece con el uso	Intervenciones manuales; costo por factura	Sin intervención manual; costo por factura menor que el del procedimiento manual y no mayor que el del bot, contando la mantención

3.1.3 RESTRICCIONES

Las restricciones son condiciones que el diseño no elige, sino que recibe. Provienen de tres fuentes: la infraestructura y las herramientas que la empresa ya utiliza, los sistemas con los que la solución debe integrarse sin modificarlos y las normas de seguridad corporativas descritas en la sección 1.7. La Tabla 14 las reúne junto con su consecuencia. Su efecto es acotar el espacio de decisiones de las secciones siguientes, el mérito del diseño no está en elegir un proveedor ni en rediseñar EMER, sino en resolver los problemas del Capítulo 1 dentro de esos márgenes.

Tabla 14: Restricciones del diseño. Fuente: Elaboración propia a partir de datos de Evol.

ID	Tipo	Restricción	Consecuencia para el diseño
R1	Infra-estructura	La empresa opera sobre Amazon Web Services, en la región us-east-1	Los servicios se seleccionan dentro de ese proveedor (sección 3.2)
R2	Integración	EMER recibe las facturas por su interfaz de carga existente, con una estructura de datos definida, y no se modifica	La extracción entrega exactamente esa estructura y la carga regula el envío hacia EMER
R3	Integración	El catálogo de conceptos se mantiene en el *backend* de EMER y se consulta por emisor	La validación lo consulta en lugar de duplicarlo
R4	Entrada	Las facturas se depositan en SharePoint, en una carpeta por formato que alimenta el analista	La captura lee esas carpetas; la obtención queda fuera
R5	Tecnológica	El equipo desarrolla en NestJS con TypeScript y despliega con Serverless Framework mediante un pipeline de integración continua	Define el lenguaje de los servicios y la herramienta de infraestructura como código
R6	Organiza-cional	Las normas de seguridad corporativas impiden enviar documentos a terceros no gobernados por la empresa y condicionan los permisos sobre las cuentas	Los modelos se ejecutan dentro de la cuenta de la empresa; el acceso a SharePoint queda sujeto a la aprobación del área de TI
R7	Operacional	Durante el desarrollo de la memoria la solución convive con la automatización actual	Cada carpeta de entrada la procesa uno solo de los dos sistemas

3.2 SELECCIÓN DE SERVICIOS

Esta sección responde al segundo objetivo específico: seleccionar, mediante criterios explícitos, el servicio de nube que implementa cada componente de la solución. Dado que el proveedor es una restricción (R1), la selección ocurre dentro de AWS y se fija por cinco criterios, derivados de los escenarios y restricciones de la sección 3.1:

- **Ajuste al atributo**. El servicio favorece el atributo de calidad que el componente debe asegurar.
- **Operación administrada**. No exige administrar servidores ni sistemas operativos (L5).
- **Costo según el uso**. Su costo sigue al volumen procesado, en lugar de ser fijo, en coherencia con la carga concentrada del *peak* mensual (E8).
- **Conocimiento del equipo**. Es compatible con las herramientas y la experiencia del área de Desarrollo y Tecnología (R5).
- **Confidencialidad**. Los documentos no salen de la cuenta de la empresa (R6, E7).

La Tabla 15 aplica esos criterios a cada componente. En los componentes con diversas alternativas, se indica cuál se eligió, el criterio que decidió la elección y el registro de decisión en que se desarrolla el razonamiento completo (sección 3.4). En otros, el proveedor ofrece un único servicio de su categoría (el almacenamiento de objetos, la gestión de permisos), o la elección viene dada por una restricción, como la herramienta de infraestructura como código, en esos casos basta una línea de justificación.

Tabla 15: Servicio seleccionado para cada componente de la solución. Fuente: Elaboración propia.

Componente	**Alternativas consideradas**	**Servicio seleccionado**	**Criterio de selección**	**Detalle**
Cómputo de los servicios	Instancia EC2; Contenedores en ECS Fargate; Funciones	AWS Lambda	Operación y costo: sin servidores y con pago por invocación	ADR-01
Detección de facturas nuevas	Notificaciones de cambio de Graph; Sondeo programado	Amazon EventBridge Scheduler con consulta *delta*	Configuración: latencia reducida por el intervalo, sin puntos expuestos	ADR-07
Almacena-miento de documentos	-	Amazon S3	Configuración: genera un evento al recibir un archivo y aplica reglas de ciclo de vida	ADR-04

Comunicación entre servicios	Invocación directa; Colas	Amazon SQS, una cola por etapa, cada una con su DLQ	Configuración: absorbe el *peak*, desacopla las etapas y retiene lo que falla	ADR-03
Coordinación del flujo	Coreografía mediante colas; Step Functions Express; Step Functions Standard	Coreografía mediante colas de SQS, sin un servicio adicional	Equipo: tecnología que el área ya opera; el estado por factura se registra en DynamoDB	ARD-03
Lectura del documento	Análisis de facturas de Textract; Lectura directa por el modelo; Obtención de texto con Textract	Amazon Textract (obtención de texto)	Configuración: admite documentos escaneados sin cambiar el flujo	ADR-02
Interpreta-ción del contenido	Servicios de terceros; Modelos en la cuenta de la empresa	Amazon Bedrock	Confidencialidad: los proveedores de modelos no acceden a los datos	ADR-02
Estado del proceso	Base de EMER; Base relacional propia; Solo S3; Base documental	Amazon DynamoDB	Configuración, opera-ción y costo: acceso por clave, sin servidor y con pago por uso	ADR-04
Consulta del estado	-	Amazon API Gateway con una función Lambda	Equipo: mismo esquema de autenticación del backend de EMER	3.5.5
Resumen diario	-	EventBridge Scheduler, Lambda y el servicio de correo de la empresa	Equipo: reutiliza el servicio de correo existente	3.5.5

Permisos	-	AWS IAM, con un rol por función	Confidencialidad: mínimo privilegio por servicio	3.6
Registros, métricas y alarmas	CloudWatch; CloudWatch con trazas distribuidas (*X-Ray*)	Amazon CloudWatch	Costo: los registros con el identificador del documento y la tabla de estado permiten reconstruir el recorrido de cada factura	3.6
Infraestructura como código	-	*Serverless Framework*	Restricción R5	3.6

La tabla muestra que la selección no es un catálogo de servicios, en cada componente se busca alternativas, el criterio decisivo proviene de un escenario de calidad o de una restricción, lo que satisface la exigencia del segundo objetivo específico. El criterio de conocimiento del equipo pesa sobre todo en los componentes secundarios (la consulta, el resumen, la infraestructura como código), donde apartarse de lo que el equipo ya utiliza no aportaría una mejora que lo justifique, y resultó decisivo en la coordinación del flujo (ADR-03).

3.3 ARQUITECTURA PROPUESTA: VISIÓN GENERAL

La arquitectura propuesta organiza el procesamiento de las facturas como un conjunto de cinco servicios independientes, ejecutados como funciones AWS Lambda en la cuenta de la empresa y coordinados mediante eventos y colas: cada servicio consume los mensajes de su propia cola de Amazon SQS y, al terminar su etapa, deja un mensaje en la cola del siguiente. Esta sección la presenta en conjunto, con los dos primeros niveles del modelo C4. Las razones de cada decisión se desarrollan en la sección 3.4, y el detalle de cada servicio, en la 3.5.

3.3.1 CONTEXTO

El diagrama de contexto (Figura 2) sitúa la solución frente a sus usuarios y a los sistemas con que se relaciona. Hacia la entrada, el analista del equipo GCP deposita las facturas en las carpetas de SharePoint, como lo hace hoy, y la solución las lee sin que medie ninguna otra acción. Hacia la salida, entrega los datos a EMER por su interfaz de carga y consulta al *backend* de EMER el catálogo de conceptos de cada emisor. Información hacia el usuario contempla: el equipo GCP recibe un resumen diario por correo, enviado mediante el servicio de correo que la empresa ya utiliza, y puede consultar el estado de cada factura. Por otro lado el área de Desarrollo y Tecnología opera y mantiene la solución. La cadena posterior a la carga, la compilación del Reporte de Validación y su envío al cliente, permanece intacta dentro de EMER.

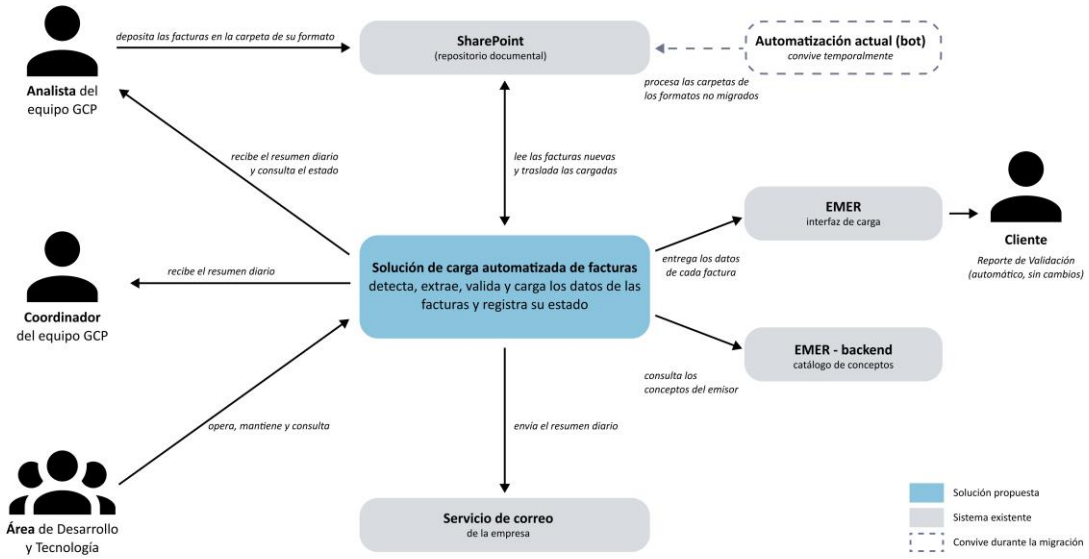


Figura 2: Diagrama de contexto de la solución (C4, nivel 1). Fuente: Elaboración propia.

3.3.2 CONTENEDORES

El nivel de contenedores (Figura 3) abre la solución en sus unidades desplegables. La descomposición sigue el criterio de etapa del flujo descrito en la sección 2.2.2, y cada etapa corresponde a una responsabilidad que cambia por razones propias:

- **Captura**. Detecta las facturas nuevas en SharePoint, las copia a Amazon S3, identifica su emisor y las registra. Una vez cargada la factura, traslada el documento a su carpeta definitiva.
- **Extracción**. Obtiene el texto del documento mediante Amazon Textract y lo interpreta con un modelo de lenguaje a través de Amazon Bedrock, que devuelve los datos en la estructura que recibe EMER.
- **Validación**. Aplica reglas determinísticas sobre ese resultado y decide si la factura puede cargarse o debe derivarse a revisión.
- **Carga**. Entrega los datos a EMER, una factura a la vez.
- **Estado**. Registra la etapa y el resultado de cada factura, responde las consultas a través de Amazon API Gateway y elabora el resumen diario.

Cada servicio es un módulo del proyecto NestJS de la empresa con su propia definición de despliegue, de modo que modificar uno no obliga a desplegar los demás, lo que responde de forma directa a L7. Los servicios no se invocan entre sí. La captura deposita el documento en S3, lo que genera un evento que queda en la cola de extracción. Desde ahí, cada servicio lee su propia cola y, al terminar, deja un mensaje en la del siguiente (extracción, validación, carga), de modo que ninguno conoce el flujo completo; cada uno registra además su avance

en la tabla de estado. Los elementos de cada etapa (el documento, el texto, el resultado de la extracción, lo enviado a EMER y su respuesta) se guardan en S3, y el estado de la factura, en Amazon DynamoDB.

3.3.3 EL RECORRIDO DE UNA FACTURA

En el caso normal, una factura recorre la arquitectura en siete pasos:

1. Cada quince minutos, Amazon EventBridge Scheduler activa la captura, que consulta los cambios ocurridos en las carpetas de entrada (Sharepoint).
2. Por cada documento nuevo, lo copia a S3 con su emisor ya identificado y lo registra como recibido. Si el emisor está marcado como de documentos compuestos, lo deriva de inmediato a revisión.
3. La escritura del documento genera un evento, que queda en la cola de extracción.
4. La extracción obtiene los datos y deja un mensaje en la cola de validación.
5. La validación los verifica. Si los aprueba, deja un mensaje en la cola de carga, si no, la factura pasa a revisión con el motivo.
6. La carga entrega los datos a EMER, lo que gatilla, como hoy, el Reporte de Validación, y deja un mensaje en la cola de traslado.
7. La carga traslada el documento a su carpeta definitiva y la factura queda registrada en el sistema.

Tres rasgos de este recorrido responden a los escenarios de la sección 3.1. La detección periódica reemplaza la espera a la ejecución diaria (E5). Como cada mensaje corresponde a una sola factura, una falla queda contenida en ella (E4), y el registro que cada servicio escribe al terminar su etapa permite conocer siempre en qué punto está (E6). Y las colas, junto con la carga de una factura a la vez, absorbe el *peak* mensual sin exigir a EMER más de lo que puede atender (E3, E8).

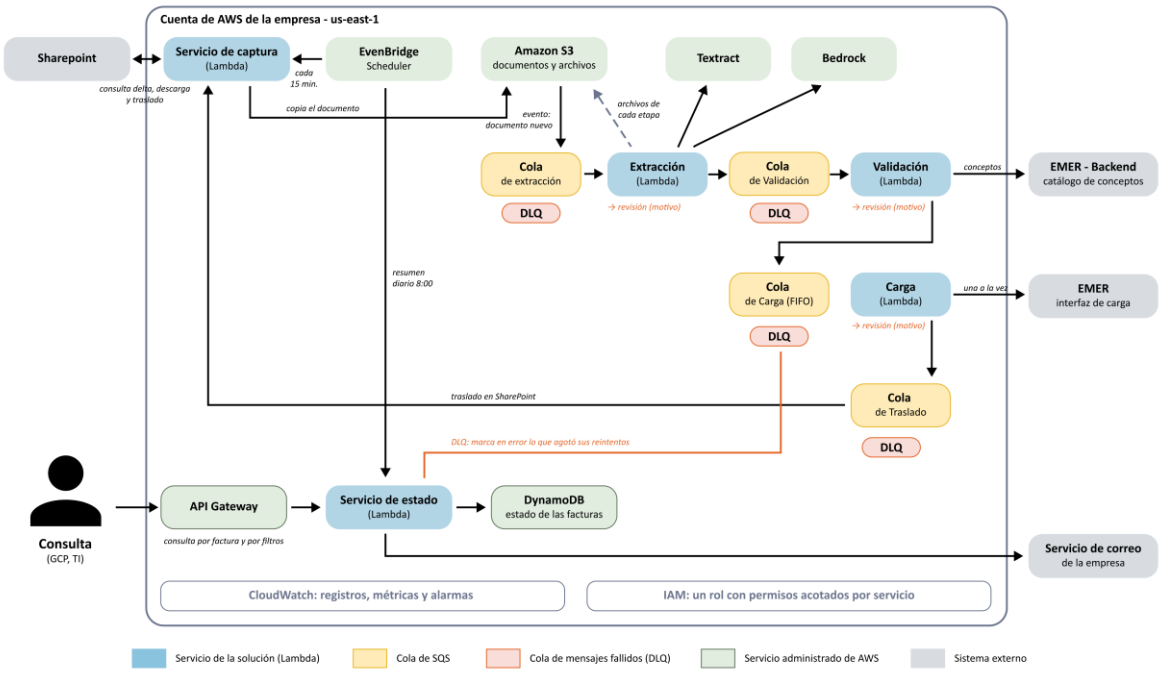


Figura 3: Diagrama de contenedores de la solución (C4, nivel 2). Fuente: Elaboración propia.

3.4 DECISIONES ARQUITECTÓNICAS

Las decisiones que dan forma a la arquitectura se documentan como registros de decisión arquitectónica (*ADR*) (sección 2.1.3). Se registran siete, las de mayor costo de revertirlas y las que responden de forma directa a los escenarios de la sección 3.1. Todas siguen el mismo formato (contexto, alternativas evaluadas, decisión y consecuencias) y declaran tanto lo que se gana como lo que se acepta de condicionante. La Tabla 16, al final de la sección, las resume.

3.4.1 ADR-01 ESTILO ARQUITECTÓNICO Y MODELO DE CÓMPUTO

- **Contexto**. La automatización actual es un proceso único alojado en la misma instancia que EMER (L3), con un costo fijo que no depende de la carga (L5) y que obliga a desplegar todo ante cualquier cambio (L7). Aunque la carga se concentra entre la segunda y la tercera semana del mes.
- **Alternativas**.
    a) Mantener un proceso único, reorganizado en módulos, en una instancia propia.
    b) Microservicios en contenedores sobre Amazon ECS con Fargate.
    c) Microservicios como funciones AWS Lambda.
- **Decisión**. Se adopta (c). La alternativa (a) separa el proceso del servidor de EMER, pero no resuelve L7 ni el costo fijo. La (b) permite ejecuciones sin límite de duración, una ventaja que este caso no necesita, porque cada etapa procesa un único documento, y añade la gestión de imágenes y de tareas en ejecución.

- **Consecuencias**. Cada servicio se despliega por separado, no hay servidores que administrar y el costo sigue al uso, lo que calza con el perfil concentrado descrito en la sección 2.3.2. A cambio, los límites de ejecución obligan a pasar los documentos por referencia (ADR-04), la arquitectura queda ligada al proveedor AWS, lo que la restricción R1 ya asume, y aparecen los costos operacionales de la Tabla 8. El arranque en frío no es relevante, nadie espera la respuesta en línea. Se cumple, además, la secuencia que Lewis y Fowler (2014) recomiendan, el bot fue el monolito inicial, y sus límites, ya medidos, justifican ahora la distribución.

3.4.2 ADR-02 ENFOQUE DE EXTRACCIÓN

- **Contexto**. Las reglas por formato fallan en cerca del 30% de las facturas (L1) y cada formato nuevo cuesta unas 25 horas (L2). Las facturas son PDF digitales, pero se desea que el flujo admita documentos escaneados sin rediseñarlo.
- **Alternativas**.
    a) Reglas por formato, como hoy.
    b) El análisis de facturas de Amazon Textract, que extrae campos con una taxonomía estándar (fecha, número, total, impuestos, ítems) (Amazon Web Services, 2026d).
    c) Entregar el PDF directamente al modelo de lenguaje.
    d) Obtener el texto con Textract e interpretarlo con un modelo de lenguaje en Amazon Bedrock, mediante una instrucción que declara la estructura de salida.
- **Decisión**. Se adopta (d), una instrucción genérica, con añadidos por formato cuando hacen falta, que devuelve exactamente la estructura que recibe EMER. El modelo, actualmente Claude Haiku 4.5, se define por configuración. La alternativa (b) se descarta porque su taxonomía, pensada para facturas comerciales, no contiene los cargos propios del sector eléctrico ni los identificadores de instalación y medidor. La (c) se evaluó y no se adoptó, simplifica el flujo, pero elimina la capa de lectura independiente del modelo.
- **Consecuencias**. La extracción pasa a depender del significado y no de la posición (sección 2.4.3), e incorporar un formato consiste en ajustar una instrucción (E2). Separar la lectura de la interpretación permite admitir escaneados y conservar el texto como archivos, de modo que la interpretación puede repetirse sin volver a leer el documento. Como *Bedrock* se ejecuta en la cuenta de la empresa y los proveedores de modelos no acceden a las instrucciones ni a las respuestas (Amazon Web Services, 2026e) , se cumple E7. A cambio, se asumen los riesgos de alucinación y de no determinismo, que obligan al ADR-06, y un costo que varía con cada documento (sección 3.6).

3.4.3 ADR-03 COORDINACIÓN DEL FLUJO

- **Contexto**. Cada factura atraviesa tres etapas. El equipo GCP necesita saber en cuál está (L6, E6), una falla no debe afectar a otras facturas (E4) y los reintentos deben distinguir los errores transitorios de los permanentes (E3). El área de Desarrollo y Tecnología, que operará la solución, trabaja habitualmente con colas y funciones, pero no ha operado servicios de orquestación de flujos.
- **Alternativas**.
    a) Coreografía: cada servicio consume los mensajes de su propia cola de Amazon SQS y, al terminar, deja un mensaje en la cola del siguiente.
    b) Orquestación con flujos Express de AWS Step Functions.
    c) Orquestación con flujos Standard de AWS Step Functions, una ejecución por factura.
- **Decisión**. Se adopta (a). La alternativa (c) fue la opción inicial del diseño, porque conserva el historial de cada ejecución y declara los reintentos de cada paso en un solo lugar (Amazon Web Services, 2026f); la (b) comparte sus inconvenientes y además no guarda historial propio. Se revisó tras la evaluación del área de Desarrollo y Tecnología, que consideró que incorporar un servicio que no ha operado agrega una complejidad que este volumen no justifica. El criterio de conocimiento del equipo (sección 3.2) resulta entonces decisivo: Lewis y Fowler (2014) advierten que el resultado de una arquitectura distribuida depende de la experiencia de quien la implementa. Las propiedades que aportaba la orquestación se obtienen por otra vía: el estado de cada factura lo registra cada servicio en la tabla de estado (ADR-04), y los reintentos los administran las colas y sus DLQ.
- **Consecuencias**. La solución usa solo servicios que el equipo ya opera, cada etapa queda desacoplada de las demás (sección 2.2.3) y desaparece el costo por transición. A cambio, la secuencia del flujo no queda escrita en un solo lugar, sino repartida en qué cola lee y en cuál escribe cada servicio; la visibilidad depende de que cada servicio registre su etapa, lo que el Capítulo 4 verifica con el escenario E6; la distinción entre errores transitorios y permanentes la aplica cada servicio (sección 3.5.6), y como las colas entregan cada mensaje al menos una vez, cada servicio debe comprobar el estado de la factura antes de actuar. La depuración se hace con los registros, filtrados por el identificador del documento, en lugar de una vista gráfica de cada ejecución.

3.4.4 ADR-04 PERSISTENCIA DEL ESTADO Y DE LOS ARCHIVOS

- **Contexto**. Se debe registrar el estado de cada factura y consultarlo por identificador y por filtros (L6). Cada factura produce además archivos de tamaño variable: el documento, el texto extraído, el resultado y la respuesta de EMER.
- **Alternativas**.
    a) Tablas nuevas en la base PostgreSQL de EMER.
    b) Una base relacional administrada propia.
    c) Solo Amazon S3.
    d) Amazon DynamoDB para el estado y S3 para los archivos, según el patrón de reclamo (*claim check*), en que la base guarda solo una referencia al archivo.
- **Decisión**. Se adopta (d). La alternativa (a) acopla la solución al sistema que la restricción R2 trata como caja negra. La (b) añade un servidor de base de datos que administrar para un acceso que es, casi siempre, por clave. La (c) no permite consultar por filtros. Un ítem de DynamoDB admite a lo más 400 KB (Amazon Web Services, 2026g), y el texto de una factura extensa puede superarlo, por eso los archivos van a S3.
- **Consecuencias**. La base guarda solo el estado del proceso, con un esquema flexible y pago por uso (sección 2.3.4), y S3 aplica reglas de ciclo de vida a los archivos temporales. A cambio, las consultas por filtro exigen índices diseñados de antemano, y como la escritura en ambos servicios no es transaccional, el estado se actualiza solo después de guardar el archivo al que apunta. Los datos de la factura no se duplican como fuente de verdad, una vez cargados, esa fuente es EMER.

3.4.5 ADR-05 CARGA EN EMER E IDEMPOTENCIA

- **Contexto**. EMER no se modifica (R2) y se desconoce cuántas cargas simultáneas tolera. Cada carga gatilla la compilación del Reporte de Validación, que se envía al cliente una vez al mes, una carga duplicada o parcial deja en EMER datos que el reporte ya enviado no refleja, y corregirlo exige una intervención manual. Las colas entregan cada mensaje al menos una vez (sección 2.2.3).
- **Alternativas**.
    a) Cargar en paralelo, a medida que las facturas se validan.
    b) Cargar por lotes, a horas fijas.
    c) Leer las facturas validadas desde una cola FIFO, que las entrega en el orden en que llegaron y de a una, de modo que la cola las retenga hasta que puedan cargarse.
- **Decisión**. Se adopta (c), en una cola FIFO, los mensajes de un mismo grupo se entregan en orden, y si uno falla se agotan sus reintentos antes de entregar el siguiente, con un único grupo, la función de carga se ejecuta de a una vez (Amazon Web Services, 2026h). La alternativa (b) volvería a introducir la espera que L4 busca eliminar. La idempotencia se asegura con una clave por factura (emisor, número de factura y periodo): antes de cargar se consulta el estado, y si la factura ya figura como cargada, el mensaje se descarta sin llamar a EMER, que opera como segunda barrera.

- **Consecuencias**. EMER recibe una carga a la vez, mientras la extracción sigue en paralelo, si la medición en QA mostrara que EMER tolera más, la cola podría repartirse en varios grupos, que se procesan en paralelo. A cambio, en el *peak* la carga se convierte en el cuello de botella del flujo. Además, mientras una factura espera sus reintentos, las siguientes esperan detrás de ella: si EMER deja de responder, toda la carga se detiene hasta que vuelve o hasta que esa factura agota sus intentos, cerca de una hora después, y una alarma avisa cuando un mensaje lleva más de una hora en espera (sección 3.6.2). Para EMER, esa detención es preferible a recibir cargas mientras está caído.

3.4.6 ADR-06 VALIDACIÓN DE LOS DATOS EXTRAÍDOS

- **Contexto**. Un modelo de lenguaje puede devolver valores plausibles pero ausentes del documento (sección 2.4.3), y un concepto no reconocido se compara contra cero en el Reporte de Validación, lo que genera una alerta que llega al cliente (P2).
- **Alternativas**.
    a) No validar y confiar en las alertas del Reporte de Validación.
    b) Un segundo modelo de lenguaje que revise la respuesta del primero.
    c) Reglas determinísticas, con derivación a revisión.
- **Decisión**. Se adopta (c), en tres capas: estructura (campos obligatorios, tipos y fechas), coherencia aritmética y de negocio (suma de cargos, periodo, duplicados) y conceptos, contrastados con el catálogo del emisor que expone el *backend* de EMER y con una lista de conceptos que no se cargan. La factura que no supera alguna capa se deriva a revisión con su motivo, nunca se descarta. La alternativa (a) deja que el error llegue al cliente. La (b) duplica el costo y la latencia, es igualmente no determinista y no elimina el riesgo, lo desplaza a otro modelo.
- **Consecuencias**. Cada rechazo es reproducible y explicable, y su costo es despreciable. A cambio, un valor mal extraído que resulte coherente con todas las reglas no se detecta, es el riesgo residual del diseño, que el Capítulo 4 mide mediante la precisión por campo.

3.4.7 ADR-07 OBTENCIÓN DESDE SHAREPOINT

- **Contexto**. La espera de hasta 24 horas (L4) debe bajar a menos de una hora (E5). SharePoint no emite eventos hacia AWS, y el acceso a sus carpetas está sujeto a las normas de seguridad corporativas (R6).
- **Alternativas**.
    - Suscribirse a las notificaciones de cambio de Microsoft Graph.
    - Consultar periódicamente los cambios mediante la consulta delta de Graph, que devuelve solo lo nuevo desde la consulta anterior (Microsoft, 2017).
- **Decisión**. Se adopta (b), cada quince minutos, implementada en NestJS. Las notificaciones exigen un punto de recepción público y suscripciones que vencen en menos de 30 días y deben renovarse, además, para archivos declaran una latencia máxima de seis horas y solo informan el identificador del elemento, por lo que

igualmente obligan a consultar (Microsoft, 2024a). Para autenticarse se propone una aplicación registrada con permiso acotado a las carpetas de entrada (Microsoft, 2024b), en lugar del usuario y contraseña que emplea hoy el *bot*, un flujo que Microsoft desaconseja (Microsoft, 2025).

- **Consecuencias**. La latencia de detección queda acotada por el intervalo, no hay componentes expuestos a internet y, si una consulta falla, la siguiente retoma desde el último punto registrado. A cambio, se consulta aunque no haya cambios, lo que tiene un costo despreciable. Dos aspectos quedan sujetos a prueba: la biblioteca de Graph para NestJS y el registro de la aplicación. Si el primero falla, la captura se implementa en Python y la solución pasa a ser políglota; si el segundo no se aprueba, se mantiene el usuario y contraseña con el token reutilizado mientras sea válido, y se asume ese riesgo.

Tabla 16: Resumen de las decisiones arquitectónicas. Fuente: Elaboración propia.

ADR	**Decisión**	**Principal alternativa descartada**	**Responde a**
ADR-01	Microservicios en AWS Lambda	Contenedores en ECS Fargate	L3, L5, L7; E2, E8
ADR-02	Textract para leer y Bedrock para interpretar, con salida en la estructura de EMER	Reglas por formato; PDF directo al modelo (evaluada, no adoptada)	L1, L2; P3, P4; E1, E2, E7
ADR-03	Coreografía: una cola de SQS por etapa; estado por factura en DynamoDB	Orquestación con Step Functions (opción inicial, revisada)	L3, L6; P6; E3, E4, E6
ADR-04	DynamoDB para el estado y S3 para los archivos	Tablas en la base de EMER	L6; E6
ADR-05	Carga de una en una: cola de carga FIFO; idempotencia por clave de factura	Carga en paralelo	P1, R2; E3
ADR-06	Validación determinística en tres capas y derivación a revisión	Segundo modelo como revisor	P2; E1
ADR-07	Sondeo cada 15 minutos con consulta delta de Graph, en NestJS	Notificaciones de cambio de Graph	L4; P6; E5, E7

3.5 DISEÑO DETALLADO DE LOS SERVICIOS

Esta sección describe cada servicio con un mismo molde (responsabilidad, disparador, entradas y salidas, y manejo de errores) y cierra con el flujo que los coordina, los estados por los que pasa una factura y el modelo de datos. En todos los servicios los errores se clasifican según la distinción de la sección 2.2.3: los transitorios devuelven el mensaje a su cola, que lo reintenta con espera, y los permanentes derivan la factura a revisión con su motivo, sin reintentar (sección 3.5.6). Cada reintento espera el doble que el anterior. En la extracción, la validación y el traslado, la espera va de 5 segundos a un máximo de 5 minutos, con cinco intentos, en la carga, de 10 segundos a 15 minutos, con diez intentos, cerca de una hora en total, porque EMER puede quedar fuera de servicio por un rato, por ejemplo durante su reinicio diario, cerca de las 4:00.

3.5.1 SERVICIO DE CAPTURA

Amazon EventBridge Scheduler lo activa cada quince minutos. Su comportamiento lo determina una tabla de configuración que asocia cada carpeta *<entorno>/cron/<formato>* con el identificador del emisor en EMER, su tipo (distribuidora o suministradora), una marca de documentos compuestos, una marca de actividad y la lista de conceptos del emisor que no se cargan en EMER, como el saldo anterior, los intereses o el redondeo. Solo se leen las carpetas activas, lo que resuelve la convivencia con el bot (R7), cada carpeta la procesa uno solo de los dos sistemas, y migrar un formato consiste en cambiar su marca.

En cada ejecución, el servicio consulta los cambios de cada carpeta activa desde el último punto registrado. Por cada archivo nuevo calcula un identificador del documento a partir de su contenido, de modo que un mismo archivo depositado dos veces se reconoce y se omite. Luego lo copia a S3, bajo *entrada/<emisor>/<AAAA-MM>/* (el mes de llegada, porque el periodo facturado solo se conoce después de extraer), con el emisor en los metadatos del objeto, y lo registra como recibido. Un documento compuesto se reconoce de dos formas antes de extraerlo: por la marca del emisor en la configuración y por el nombre del archivo, porque los analistas nombran la segunda parte con el sufijo “SP” o “PE”, la misma convención que usa el *bot*. En ambos casos, el archivo se copia directamente a *revision/*, queda registrado con su motivo y no se extrae.

Su segunda responsabilidad es ordenar los documentos en SharePoint según cómo terminó su procesamiento: los cargados y los omitidos por duplicados pasan a la carpeta de salida del mes y tipo de emisor que corresponde, y los que quedan en revisión o en error, a una carpeta REVISION o ERROR del emisor. Así, el equipo GCP ve el resultado en el mismo repositorio en que trabaja, y devolver un documento desde esas carpetas a la de su emisor es otra forma de pedir su reproceso. Si Graph no responde, la ejecución termina y la siguiente retoma desde el último punto. Si ese punto caduca, Graph obliga a recorrer de nuevo la carpeta completa (Microsoft, 2017), y el identificador por contenido evita que los archivos ya procesados se registren otra vez.

3.5.2 SERVICIOS DE EXTRACCIÓN

Lee de la cola de extracción la referencia al documento en S3 y trabaja en dos pasos. Primero obtiene el texto con Textract y lo guarda en *procesamiento/<documentoId>/.* Luego construye la solicitud al modelo en Bedrock, compuesta por una instrucción genérica, el añadido del formato cuando existe y el texto del documento, y recibe un objeto JSON con la estructura que EMER espera. Esa estructura agrupa la identificación del documento (emisor, folio, titular y tipo), el suministro (códigos de instalación y de medidor), las fechas (emisión, vencimiento, inicio y término del periodo), los totales (neto y con exentos) y la lista de cargos, cada uno con su concepto, consumo y costo.

El servicio verifica solo que la respuesta sea un JSON con esa forma, la corrección de su contenido corresponde a la validación. Guarda el resultado junto con los *tokens* consumidos y el costo estimado, y deja un mensaje en la cola de validación. La instrucción y sus añadidos por formato se versionan con el código del servicio, de modo que incorporar o ajustar un formato exige desplegar solo este servicio (E2); por tratarse de información sensible de la empresa, su texto no se muestra en este documento.

Las restricciones de tasa y los tiempos de espera agotados de Textract o Bedrock son errores transitorios. Un documento dañado, sin texto o de más de diez páginas es un error permanente, ninguna factura real supera ese largo, y el límite protege de archivos equivocados o maliciosos. Una respuesta mal formada se reintenta una vez, porque el modelo no es determinista (sección 2.4.3), y si persiste, la factura pasa a revisión.

3.5.3 SERVICIO DE VALIDACIÓN

Aplica las tres capas del ADR-06 sobre el resultado de la extracción (Tabla 17). Es también el primer punto en que se conoce la clave de la factura (emisor, folio y periodo), que se registra para las comprobaciones de duplicado y para la idempotencia de la carga. La consulta del catálogo de conceptos se hace al *backend* de EMER con el identificador del emisor, y devuelve los conceptos de ese emisor, el concepto general al que pertenece cada uno y si es obligatorio. Si el catálogo no responde, el error es transitorio. Toda factura que no supera una capa se copia a *revision/<documentoId>/* y queda en revisión con su motivo. La que las supera pasa a la cola de carga.

No todos los cargos de una factura se cargan en EMER, algunos, como el saldo anterior, los intereses o el redondeo, no intervienen en la validación del cobro. Por eso la extracción entrega todos los cargos del documento, y es la validación la que decide cuáles se cargan, clasificando cada uno en tres grupos: reconocido, si figura en el catálogo del emisor, y se carga; excluido, si figura en la lista de conceptos que no se cargan de la tabla de configuración, y se omite, aunque queda registrado en el resultado de la extracción; y desconocido, si no figura en ninguna de las dos, en cuyo caso la factura se deriva a revisión con el nombre y el monto del cargo. Así, un cargo relevante que falte en el catálogo nunca se descarta en silencio. Al revisar uno que sea desconocido, el analista lo agrega al catálogo o a la lista de excluidos, de modo que las facturas siguientes de ese emisor ya no se detienen

por él. La coherencia aritmética se verifica sobre todos los cargos extraídos, incluidos los excluidos, porque lo que comprueba es que la extracción esté completa. Como mejora futura, la marca de obligatorio del catálogo de EMER podría convertirse en una categoría (obligatorio, opcional o excluido) para mantener ambas listas en un solo lugar. Enumerando las capas se tiene, 1. Estructura, 2. Coherencia, 3. Conceptos.

Tabla 17: Reglas del servicio de validación. Fuente: Elaboración propia.

Capa	Verificación	Resultado si no se cumple
1	Campos obligatorios presentes; valores numéricos y fechas válidos; inicio del periodo anterior o igual a su término	Revisión: estructura inválida
2	Suma de los costos de los cargos igual al total neto, dentro de una tolerancia.	Revisión: descuadre de montos
2	Consumos y costos dentro de los signos esperados (puede haber caso especial de reliquidación)	Revisión: valor fuera de rango
2	La clave de la factura ya figura como cargada	Omitida por duplicada, sin revisión
2	Existe otra factura del mismo emisor, instalación y periodo con distinto folio	Revisión: posible documento compuesto
3	Cada cargo extraído figura en el catálogo del emisor o en su lista de conceptos que no se cargan; los segundos se omiten de la carga	Revisión: concepto no reconocido, con su nombre y monto
3	Todos los conceptos obligatorios del emisor están presentes	Revisión: concepto obligatorio ausente

3.5.4 SERVICIO DE CARGA

Lee de la cola de carga una factura validada, comprueba por su clave que no haya sido cargada antes (ADR-05) y la envía a la interfaz de carga de EMER. Guarda en S3 lo enviado, lo recibido y, si la carga resulta, deja un mensaje en la cola de traslado. Lee una cola FIFO, que le entrega las facturas en orden y de a una. La falta de respuesta, el tiempo agotado o un error del servidor son errores transitorios, un rechazo de EMER por el contenido de los datos es permanente y deriva la factura a revisión. Si EMER cierra la sesión, el servicio inicia sesión otra vez y reintenta de inmediato, si rechaza las credenciales, la factura pasa a error sin reintentar, porque esperar no lo resuelve.

3.5.5 SERVICIO DE ESTADO

Mantiene el registro de cada factura en DynamoDB, que cada servicio actualiza al terminar su etapa. Además, lee las *DLQ* de todas las colas y registra como en error las facturas cuyos mensajes agotaron sus reintentos. Ofrece tres operaciones:

1. Consulta, expuesta mediante API Gateway, por identificador de documento o de factura, y por filtros de emisor, periodo y estado. Se protege con el mismo mecanismo que usa el *backend* de EMER (un token con los permisos del usuario), lo que permitiría, más adelante, consultarla desde la interfaz de EMER. Esa vista queda fuera del alcance.
2. Reprocesamiento de una factura en revisión o en error, que deja nuevamente el documento ya almacenado en la cola de extracción, sin volver a depositarlo (RF10). También se pide devolviendo el documento desde la carpeta *REVISION/* o *ERROR/* a la de su emisor.
3. Resumen diario, activado por el programador de tareas todos los días a las 8:00 y enviado al equipo GCP mediante el servicio de correo existente de la empresa. Contiene una tabla con las facturas del día y del mes por estado y por tipo de emisor, y la lista de las que requieren acción (en revisión o en error) con su emisor, folio y motivo.

3.5.6 FLUJO, ESTADOS Y MODELO DE DATOS

El flujo entre colas que coordina los servicios anteriores es el que muestra la Figura 3. Todos aplican la misma política de errores. Ante un error permanente, el servicio registra el motivo, copia el documento a *revision/* y descarta el mensaje. Ante uno transitorio, deja que el mensaje vuelva a su cola, que lo reintenta con espera hasta un máximo de recepciones, agotado ese máximo, el mensaje pasa a la DLQ de la cola y el servicio de estado registra la factura en error. Como esta política no queda escrita en un solo lugar, se implementa en una biblioteca común que cada servicio incorpora. Y como cada cola entrega los mensajes al menos una vez, cada servicio consulta el estado de la factura antes de actuar y descarta el trabajo ya hecho. El traslado en SharePoint es la última etapa, y su falla no invalida una carga ya realizada: se registra y se reintenta en la siguiente ejecución de la captura. Del mismo modo, si un mensaje llega a una DLQ después de que EMER confirmó la carga, la factura queda como cargada y no en error.

En su ciclo de vida, una factura pasa por los estados recibida, en extracción, extraída, validada y cargada, y desde cualquiera de las etapas intermedias puede pasar a revisión, con su motivo, o a error, si agotó sus reintentos. Los estados finales son dos: cargada y omitida por duplicada. Los estados de revisión y de error no son finales, porque admiten reproceso.

En el modelo de datos, la tabla principal se indexa por el identificador del documento, que existe desde la captura, y tiene índices secundarios por clave de factura, para la idempotencia, por estado y fecha, para los filtros y el resumen diario, y por emisor,

instalación y periodo, para detectar documentos compuestos no marcados. La tabla de configuración guarda, por cada carpeta, el emisor, sus marcas y el último punto de consulta de Graph. En S3 los archivos se organizan por documento y no por ejecución.

3.6 ASPECTOS TRANVERSALES

Esta sección reúne cuatro aspectos que no pertenecen a un servicio en particular, sino a la solución completa: la seguridad, la observabilidad, el despliegue y el costo.

3.6.1 SEGURIDAD

Cada función se ejecuta con un rol de IAM propio, que solo concede los permisos que su servicio necesita. La extracción, por ejemplo, puede leer los documentos de entrada (S3), escribir su resultado (DynamoDB) e invocar Textract y Bedrock, pero no puede llamar a EMER, el servicio de carga es el único que dispone de las credenciales de la interfaz de carga. De este modo, una falla o un error de programación en un servicio no compromete los recursos de los demás.

La confidencialidad exigida por E7 se sostiene en dos hechos: los documentos solo circulan entre SharePoint y la cuenta de AWS de la empresa, y los proveedores de modelos no acceden a las instrucciones ni a las respuestas procesadas por Bedrock (Amazon Web Services, 2026e) . El almacenamiento en S3 se configura sin acceso público y con cifrado en reposo. Las credenciales de Graph y de EMER se entregan a cada función como variables de entorno, definidas en el *pipeline* de integración continua y nunca escritas en el código. Esa práctica, que es la vigente en la empresa, tiene una consecuencia que debe declararse: quien tenga permisos sobre la función puede ver esas variables en la consola. Trasladarlas a un servicio de gestión de secretos queda como mejora futura. La consulta del estado se protege con el *token* del *backend* de EMER (sección 3.5.5), y el acceso a SharePoint, con una aplicación registrada de permisos acotados (ADR-07); mientras esa aplicación no se apruebe, la captura se autentica con usuario y contraseña, similar a la forma del *bot*.

3.6.2 OBSERVABILIDAD

La solución ofrece tres fuentes de información sobre su operación. La primera son los registros de Amazon CloudWatch de cada función, que incluyen el identificador del documento en cada entrada, con él se reconstruye el recorrido completo de una factura a través de los servicios. La segunda es la tabla de estado, que muestra en qué etapa está cada factura y, si se detuvo, el motivo, junto con las métricas de las colas: mensajes en espera y mensajes en las DLQ. La tercera son métricas propias: facturas procesadas, derivadas a revisión y en error, tiempo entre el depósito y la carga, y costo por factura. Estas métricas son las mismas con que el Capítulo 4 verifica los escenarios E5 y E8. Seis alarmas notifican por correo al área de Desarrollo y Tecnología, al activarse y al volver a la normalidad: cuando tres o más facturas quedan en error en una hora o superan el 10% de las recibidas; cuando un mensaje lleva más de 30 minutos esperando en la cola de extracción, de validación o de traslado, o más de una hora en la de carga; cuando un mensaje permanece más de 15 minutos en una DLQ; cuando llega a una DLQ un mensaje

que no corresponde a ninguna factura, y cuando Microsoft rechaza las credenciales de la captura, caso en que esta se suspende dos horas para no bloquear la cuenta. Así, la solución informa a dos audiencias distintas: el resumen diario se dirige al equipo GCP, y la alarma y los registros, al área técnica.

3.6.3 DESPLIEGUE

Cada servicio es un módulo del proyecto NestJS con su propio archivo de configuración de Serverless Framework, que declara sus funciones, sus permisos y sus variables. Los recursos compartidos (el bucket, las colas con sus DLQ y las tablas) se declaran en ese mismo formato, como plantillas de AWS CloudFormation, junto con la regla de ciclo de vida de S3. El despliegue lo ejecuta el pipeline de integración continua de la empresa. Aunque la consola de AWS permite crear recursos a mano, el diseño exige declararlos todos como código, así el ambiente de calidad (*QA*) en que se valida la solución y el de producción se construyen a partir de la misma definición, y cualquier cambio en la infraestructura queda versionado junto con el código (L7).

3.6.4 COSTO ESTIMADO

La Tabla 18 estima el costo mensual de operación para 800 facturas, con las tarifas de la región us-east-1 consultadas en septiembre de 2026. El costo de la extracción proviene de la medición realizada durante el desarrollo, el de los demás servicios se calcula a partir de las tarifas publicadas y del número de operaciones que exige cada factura.

Tabla 18: Costo mensual estimado de operación para 800 facturas. Fuente: Elaboración propia a partir de las tarifas publicadas por Amazon Web Services.

Servicio	**Base del cálculo**	**Tarifa**	**Costo estimado (USD/mes)**
Textract y Bedrock	Medido: USD 0,02 por factura	-	~16
SQS	~4 mensajes por factura (uno por cola), más reintentos; cada mensaje implica unas pocas solicitudes	1 millón de solicitudes gratuitas al mes	~0
Lambda	~2.900 invocaciones de la captura más unas 4 por factura	1 millón de solicitudes y 400.000 GB-s gratuitos al mes	~0
DynamoDB	~8 escrituras por factura, es decir, ~6.400	USD 0,625 por millón de escrituras	~0,01

S3	~5 archivos por factura, es decir, ~4.000 escrituras, más el volumen almacenado	USD 0,005 por 1.000 escrituras; USD 0,023 por GB-mes	~0,02 más 0,023 por GB
API Gateway, CloudWatch y EventBridge Scheduler	Consultas, registros y activaciones	USD 1 por millón de llamadas; USD 0,5 por GB-mes	~1
Total			~17

El costo queda marcado por la extracción, los demás servicios cuya tarifa ya se consultó suman centavos de dólar al mes, porque el volumen de la empresa es bajo en relación con las capas gratuitas y las tarifas por operación. A diferencia de la instancia del bot, cuyo costo de 120 a 130 dólares mensuales es fijo y compartido con otros componentes, este costo es variable, crece con el número de facturas y con su extensión, y es nulo en los días sin recepción. La comparación completa, que debe incluir las horas-persona de operación y de mantención, se realiza en el Capítulo 4.

3.7 ALCANCE DE LA IMPLEMENTACIÓN

El tercer objetivo específico se cumple con el diseño descrito en este capítulo, el cuarto exige, además, implementarlo. Como el Capítulo 4 solo puede validar lo que efectivamente se construye, esta sección precisa el estado en que queda cada componente (Tabla 19).

Tabla 19: Estado de cada componente al cierre de la memoria. Fuente: Elaboración propia.

Componente	**Estado**	**Se valida en el Capítulo 4**
Extracción	Implementado	Sí
Validación	Implementado	Sí
Carga en EMER	Implementado	Sí
Estado: consulta, reproceso y resumen diario	Implementado	Sí
Coordinación del flujo (colas SQS)	Implementado	Sí
Captura desde SharePoint	Implementado	Sí
Traslado a la carpeta definitiva	Implementado	Sí

Consulta del estado desde la interfaz de EMER	Fuera del alcance	No
Agregación de documentos compuestos	Diseñado como trabajo futuro	No; se valida solo su detección

La implementación cubre los emisores seleccionados según el criterio de la sección 1.9: primero los de mayor número de facturas mensuales, tanto distribuidoras como suministradoras, excluidos los que presentan casos especiales de entrada. Para resguardar la información de la empresa, en este documento se identifican con un código: son cuatro, DX-1 y DX-2, dos de las distribuidoras de mayor volumen, y SX-1 y SX-2, dos de las suministradoras con más facturas. Los formatos restantes siguen en la automatización actual y se incorporan con el mismo mecanismo, cambiando su marca en la tabla de configuración, el costo de hacerlo se mide en el Capítulo 4.

El mecanismo de agregación que permitiría procesar las facturas compuestas por más de un documento queda diseñado, no implementado. Su base ya existe en la arquitectura, el índice por emisor, instalación y periodo permite reunir los documentos de un mismo cobro. Se agregaría un conjunto de carga, con identificador propio, del que cada documento es una parte, y la tabla de configuración indicaría cuántas partes espera cada emisor. Cada documento validado quedaría retenido, en lugar de pasar a la cola de carga, hasta que el conjunto estuviera completo, entonces, los datos se combinarían en una sola estructura y se cargarían una única vez, de modo que la clave de idempotencia pasaría a ser la del conjunto. Si el conjunto no se completa dentro de un plazo, se derivaría a revisión. Mientras ese mecanismo no exista, la solución detecta estos casos y los deriva a revisión (sección 3.5.3), lo que asegura que ningún cobro se cargue incompleto.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

4.1 DISEÑO DE LA VALIDACIÓN

El quinto objetivo específico exige validar la solución comparándola con el procedimiento manual y con la automatización actual, mediante cuatro métricas: precisión de extracción, tiempo de procesamiento, esfuerzo de mantención y costo operacional. Este capítulo lo cumple verificando los ocho escenarios de calidad de la Tabla 13 contra sus metas. Las cuatro métricas del objetivo corresponden a los escenarios E1, E5, E2 y E8, respectivamente, los cuatro restantes (tolerancia a fallos, aislamiento, mantenibilidad y confidencialidad) se verifican porque forman parte de lo comprometido en el Capítulo 3. Solo se valida lo que la Tabla 19 declara implementado. Esta sección fija el ambiente, el conjunto de prueba y la referencia contra la que se mide, que son comunes a todo el capítulo.

4.1.1 AMBIENTE

La validación se realiza en el ambiente de calidad (*QA*) de la empresa, en la misma cuenta de AWS y en la misma región que producción. La solución se despliega desde la misma definición de infraestructura como código (*IaaC*) que se usaría en producción (sección 3.6.3), de modo que lo observado corresponde a lo que efectivamente se desplegaría. La carga se dirige a la instancia de EMER de ese ambiente, que ejecuta la cadena completa (incluida la compilación y el envío del Reporte de Validación) pero dirige los reportes a una casilla de pruebas y no a los clientes. Así se procesan facturas reales sin alterar los datos de producción y comprobando además que la carga gatilla los pasos siguientes, tal como en la operación normal.

4.1.2 CONJUNTO DE PRUEBA

El conjunto de prueba es una muestra de 100 facturas de los cuatro emisores implementados (dos distribuidoras y dos suministradoras, entre las de mayor volumen), tomadas de dos meses ya cerrados: 50 de febrero y 50 de marzo de 2026 (Tabla 20). Las facturas de cada emisor se eligieron al azar entre las recibidas en cada mes, y las distribuidoras pesan más en la muestra, como en el volumen real. Incluir dos meses permite observar si algún emisor modificó el diseño de su factura entre un periodo y otro, que es precisamente el estímulo del escenario E1. El tamaño de la muestra coincide, además, con el de un día de alta recepción, por lo que el mismo conjunto sirve para la prueba de *peak* (sección 4.3.3).

Las mismas facturas se procesan con la solución y con el *bot*, que se vuelve a ejecutar sobre el conjunto, así, ambas mediciones responden a los mismos documentos. Cada sistema lee su propia carpeta, deja los resultados en carpetas separadas y carga en su propia instancia de EMER de calidad, de modo que una ejecución no altera la otra y lo cargado por cada uno puede compararse por separado con la referencia.

Tabla 20: Conjunto de prueba por emisor. Fuente: Elaboración propia a partir de datos de Evol.

Emisor	Tipo	Febrero-2026	Marzo-2026	Total
DX-1	Distribuidora	15	15	30
DX-2	Distribuidora	15	15	30
SX-1	Suministradora	10	10	20
SX-2	Suministradora	10	10	20
Total	**-**	**50**	**50**	**100**

4.1.3 VALOR DE REFERENCIA

Medir la precisión exige conocer, para cada factura, el valor correcto de cada campo (en inglés, *ground truth*). Como referencia se toman los datos registrados en EMER de producción, que el equipo GCP ya revisó en su ciclo mensual; así se evita transcribir de nuevo cientos de facturas.

Esa referencia, sin embargo, no es infalible, sus datos los cargó una persona o el bot. Por eso se aplica una regla. Cuando la solución y el bot coinciden en un valor distinto del registrado en EMER, se revisa el documento original y, si el error estaba en EMER, se corrige la referencia, porque es poco probable que dos extracciones independientes cometan el mismo error. El número de correcciones se informa en la sección 4.2 como un indicio (no como una medida) de la tasa de error de la carga.

4.1.4 REGISTRO DE LAS MEDICIONES

No se construye ninguna herramienta adicional. La solución ya registra lo que la validación necesita (sección 3.6.2): la tabla de estado guarda la etapa, las marcas de tiempo, los tokens y el costo estimado de cada factura, los registros de CloudWatch, filtrados por el identificador del documento, muestran su recorrido por los servicios, y las métricas de las colas muestran los mensajes en espera, las recepciones de cada uno y los que llegaron a una DLQ. Cada dato de este capítulo se identifica según su origen, medido en el ambiente de calidad (*QA*), estimado a partir de tarifas o cálculos, o dato de la empresa, tomado de la línea base del Capítulo 1.

4.2 PRECISIÓN DE LA EXTRACCIÓN

Esta sección valida el escenario E1 y corresponde a la primera métrica del quinto objetivo específico. Su meta tiene dos partes (Tabla 13): una relativa, que la solución no sea menos precisa que el *bot*, y una absoluta, que ningún dato llegue a EMER sin haber superado la validación.

4.2.1 MÉTODO

La unidad de comparación son los atributos. Se comparan todos los atributos de la estructura que recibe EMER, agrupados como en la sección 3.5.2: identificación, suministro, fechas, totales y cargos. Un cargo se considera correcto cuando su concepto corresponde al mismo concepto del catálogo del emisor y su costo coinciden con la referencia, un cargo que EMER no registra por no tener concepto asociado cuenta como faltante. Los valores numéricos y las fechas se comparan después de normalizar su formato. Un monto se considera correcto cuando tiene diferencia en a lo más un 1% a la referencia, con un máximo de 100.000 pesos. La tolerancia no es nula porque los cargos de un emisor varían entre periodos (aparecen, desaparecen o cambian de nombre), lo que produce diferencias menores al compararlos con la referencia. Su valor se fijó de forma experimental, a partir de los montos observados: en una factura de cientos de miles de pesos equivale a diferencias de algunos miles, un rango que se considera razonable, porque no altera de forma relevante el control de lo que paga el cliente. Se espera reducirla a medida que se conozca mejor el comportamiento de cada emisor.

La precisión se informa con dos cifras, porque ninguno de los dos sistemas procesa todas las facturas, el *bot* falla en una parte de ellas y la solución deriva otras a revisión. La primera cifra se calcula sobre las facturas procesadas y mide la calidad de lo que cada sistema entrega a EMER. La segunda se calcula sobre el total del conjunto, contando como incorrectos los campos de las facturas no procesadas, y mide cuánto del trabajo resuelve cada uno sin intervención. Informar solo la primera favorecería al sistema que rechaza más, informar solo la segunda ocultaría que una factura derivada a revisión no es un error que llegue al cliente.

Se reportan, además:

- El porcentaje de facturas con todos sus campos correctos.
- El porcentaje de facturas derivadas a revisión y sus motivos, que no son errores, pero sí trabajo manual y se retoman en la sección 4.4.
- Los tipos de error: campo faltante, valor mal leído, concepto mal asignado y valor inexistente en el documento, este último propio de un modelo de lenguaje (sección 2.4.3).
- Las facturas con algún campo incorrecto que igualmente superaron la validación.

Esta última cifra requiere una aclaración. La parte absoluta de la meta se cumple por construcción, el flujo no permite cargar una factura que no haya pasado por la validación (sección 3.5.6). Lo que sí debe medirse es el riesgo residual declarado en el ADR-06, un valor mal extraído que resulta coherente con todas las reglas y, por tanto, no se detecta.

4.2.2 RESULTADOS

La Tabla 21 presenta la precisión por grupo de atributos y la Tabla 22, la distribución de los errores.

Tabla 21: Precisión de la extracción por grupo de campos. Fuente: Elaboración propia.

Grupo de atributos	***Bot*, sobre procesadas**	***Bot*, sobre el total**	**Solución, sobre procesadas**	**Solución, sobre el total**
Identificación	97%	71%	100%	91%
Suministro	90%	66%	98%	89%
Fechas	85%	62%	100%	91%
Totales	95%	69%	100%	91%
Cargos	90%	66%	95%	86%
Todos los atributos	91%	66%	98%	89%
Facturas todo correcto	55%	40%	88%	80%
Facturas no procesadas	-	27 de 100	-	9 de 100

Tabla 22: Errores de extracción por tipo en facturas procesadas. Fuente: Elaboración propia.

Tipo de error	***Bot***	**Solución**
Campo faltante	45	12
Valor mal leído	60	16
Concepto mal asignado	12	11
Valor inventado	-	0
Errores que superan la validación	-	39

4.3 ATRIBUTOS DE CALIDAD

Esta sección valida los escenarios E2 a E7. Para cada uno se indica el estímulo aplicado, lo que se midió y el resultado frente a la meta y a las dos líneas base.

4.3.1 MANTENIBILIDAD: NUEVO FORMATO

El escenario E2 corresponde a la métrica de esfuerzo de mantención. Se evalúa mediante un caso controlado: un emisor cuyas facturas hoy procesa solo el ***bot***, DX-3, se incorpora a la solución. Durante el trabajo se lleva una bitácora con el tiempo dedicado a cada actividad: analizar el formato, redactar el añadido de la instrucción si hace falta, registrar la carpeta en la tabla de configuración, probar con facturas del emisor y desplegar, y los servicios modificados se cuentan a partir del historial de cambios del repositorio.

La referencia del *bot* son las cerca de 25 horas que el área de Desarrollo y Tecnología estima para un extractor nuevo (Tabla 4). Esa cifra es una estimación y no una medición en las mismas condiciones, lo que se declara como límite de la comparación. En el procedimiento manual no hay desarrollo, pero el analista debe aprender a leer cada formato, un esfuerzo que no se ha cuantificado.

4.3.2 FIABILIDAD: RESPUESTAS A FALLAS

Los escenarios E3 y E4 se validan provocando fallas controladas en medio de un lote de 30 facturas (Tabla 23). Para cada falla se comprueba la respuesta esperada según el diseño (sección 3.5) y se mide:

- Facturas perdidas: documentos depositados que no tienen un estado registrado al terminar la prueba (meta de E3: ninguna);
- Facturas afectadas por una falla ajena: facturas del lote que no terminaron normalmente por una falla ocurrida en otra (meta de E4: ninguna).

El comportamiento del *bot* ante las mismas fallas no se reproduce, sino que se describe a partir de su funcionamiento, informado por el área de Desarrollo y Tecnología.

Tabla 23: Respuesta de la solución y del bot ante fallas. Fuente: Elaboración propia a partir de datos de Evol.

Falla	**Cómo se provoca**	**Respuesta esperada (Solución)**	**Observado**	**Comportamiento *bot***
EMER no responde	La dirección de carga apunta a un destino sin respuesta	El mensaje vuelve a la cola de carga y se reintenta con espera. Al restituir EMER, la factura se carga sola si no se agotó el máximo de recepciones; si se agotó, pasa a la DLQ, queda en error con su motivo.	Reintenta con espera creciente ~50 min hasta quedar en ERROR con alerta.	Falla la factura y continúa con la siguiente; no reintenta; se reprocesa volviendo a depositar el documento en el repositorio

Documento ilegible	Se deposita un PDF dañado	Error permanente; revisión con su motivo, sin reintentos	Va a revisión sin procesar.	Falla la factura y continúa; queda en el conteo del correo de errores
Concepto desconocido	Se retira un concepto que sí se considera del catálogo del emisor en *QA*	Revisión antes de cargar, con el nombre y el monto del cargo	Va a revisión, no se carga.	No lo detecta: solo verifica los conceptos obligatorios. EMER omite el cargo sin concepto, que se compara contra cero y genera una alerta al cliente
Concepto excluido	Se procesa una factura con un cargo de la lista de conceptos que no se cargan, como el saldo anterior	Se omite de la carga, queda registrado y la factura no va a revisión	Se carga sin ese cargo.	Lo envía a EMER, que lo omite por no tener concepto asociado; el resultado es el mismo
Concepto obligatorio ausente	Se marca como obligatorio en el catálogo de QA un concepto que la factura no trae	Revisión con motivo "concepto obligatorio ausente"	Va a revisión, no se carga.	Falla la factura: el bot sí verifica los obligatorios; el resultado es el mismo
Documento repetido	Se deposita dos veces el mismo PDF	Se omite en la captura por su identificador; si ya estaba cargado, se omite por duplicado	La copia se lleva a revisión. Se carga una sola vez.	No lo detecta; EMER compara los cargos y omite la carga si son iguales

Mensaje repetido	Se reenvía a la cola de validación o de carga un mensaje ya procesado	El servicio consulta el estado de la factura y descarta el mensaje, sin repetir el trabajo ni llamar a EMER	Se descarta. EMER no recibe nada de nuevo.	No usa colas
Documento compuesto	Se deposita una factura de un emisor marcado como de documentos compuestos	Revisión sin extraer	Va directo a revisión sin procesar.	Combina los dos archivos si el segundo sigue la convención de nombre interna; si no lo encuentra, carga solo el principal

La comparación muestra tres diferencias. La primera es la respuesta ante fallas transitorias, el *bot* no reintenta, y si EMER deja de responder, todo el lote falla y el analista debe volver a depositar cada documento; la solución reintenta y, si no lo consigue, conserva la factura en error lista para reprocesarse. La segunda es la detección previa, un concepto desconocido, que en el *bot* lleva al cliente como alerta (P2), en la solución se detiene antes de la carga, mientras que los conceptos excluidos siguen sin generar revisión; ante un concepto obligatorio ausente, en cambio, ambos sistemas responden igual. La tercera juega en contra de la solución: el *bot* procesa automáticamente las facturas de dos archivos que siguen su convención de nombre, mientras que la solución las deriva a revisión, porque quedaron fuera del alcance (sección 1.9). Lo que se gana a cambio es que ningún cobro se carga incompleto, cosa que el *bot* sí hace cuando no encuentra el segundo archivo. En ambos sistemas, la falla de una factura no detiene a las demás. La diferencia, en este punto, está en las fallas del proceso completo, una caída del servidor detiene el lote del *bot* hasta la siguiente ejecución programada, mientras que en la solución cada factura viaja como un mensaje independiente por las colas (sección 3.5.6).

4.3.3 DESEMPEÑO: LATENCIA Y *PEAK* MENSUAL

El escenario E5 corresponde a la métrica de tiempo de procesamiento y se mide en dos condiciones.

Fuera del *peak*. Se depositan 20 facturas de a una, en distintos momentos, y se mide el tiempo entre el depósito y su registro como cargada. Si la captura se simula depositando en S3, el tiempo se mide desde ese depósito y se le suma el intervalo de sondeo de quince minutos, que es su peor caso. Resultado: 20s y 15min con 20s: tiempo mediano y máximo, frente a una meta de a lo más una hora. En el *bot*, la espera llega a 24 horas, porque el documento se procesa recién en la ejecución nocturna, y en el procedimiento manual el ciclo completo toma cerca de dos semanas.

En el *peak*. Se depositan de una vez las 100 facturas del conjunto de prueba, un volumen similar al de los días de mayor recepción, según el equipo GCP, en esos días llegan del orden de cien facturas, porque ni siquiera los emisores más grandes entregan todas las suyas el mismo día. Se mide el tiempo total, las facturas procesadas por hora, el tiempo de extracción por factura, el tiempo de cada llamada de carga a EMER y las intervenciones manuales requeridas. Resultado: 35 min, frente a una meta de a lo más ocho horas, es decir, dentro de la misma jornada (Tabla 13). La meta responde a la necesidad de la operación (disponer del dato el mismo día) y no a la velocidad del *bot*, que procesaría ese volumen en unos 50 minutos de ejecución, pero recién en la ejecución nocturna (valor derivado).

Dos tiempos se informan por separado, porque responden a preguntas distintas. El de extracción por factura, que incluye la lectura con Textract y la interpretación del modelo, fue de unos 5 segundos con la función ya activa y de unos 10 con un arranque en frío, frente a los 30 segundos por factura del *bot*; por lo tanto, este no es un resultado que no mejore, y además las facturas se extraen en paralelo. Los otros son el tiempo hasta completar la extracción de todo el conjunto, que fue de 10 a 25 minutos desde su depósito, según el momento en que la captura detectó los archivos (consulta SharePoint 15 minutos), y el tiempo de carga, de unos 7 minutos para las 91 facturas cargadas desde que esta comenzó, es decir, unos 4,6 segundos por factura (valor derivado), cifra que incluye el tiempo entre una carga y la siguiente y, por lo tanto, acota por arriba el de cada llamada a EMER. La carga de una factura a la vez (ADR-05) era el cuello de botella previsto, con este volumen, su duración fue menor que la de la extracción del conjunto, por lo que no fue el factor que más pesó en el tiempo total.

4.3.4 VISIBILIDAD Y CONFIDENCIALIDAD

Visibilidad (E6). Se cuenta qué porcentaje de las facturas del conjunto y de las pruebas de falla tiene estado, etapa y, cuando corresponde, motivo consultables, y se verifican las consultas por identificador y por filtros de emisor, periodo y estado. Se verifica además que toda factura cuyo mensaje terminó en una DLQ figure en error en la tabla de estado, y que al cierre de cada prueba las DLQ estén vacías o reflejadas en el estado, la evidencia es la consulta de una factura con reintentos y las métricas de las colas. Se comprueba, además, que cada documento termine en la carpeta de SharePoint que corresponde a su estado. Se comprueba también que el resumen diario se generó durante las pruebas. Resultado: 100% (N de N facturas con estado, etapa y, cuando corresponde, motivo consultables), frente a una meta del 100%. El *bot* informa por correo el total de errores con un detalle adjunto, pero no registra las cargas exitosas ni permite consultar la etapa de una factura (L6); el procedimiento manual no deja registro.

Confidencialidad (E7). No se valida con una prueba, sino con una revisión de la configuración, a partir de las definiciones de despliegue y de los roles de IAM, se enumeran los servicios que reciben el documento o su contenido (SharePoint, S3, Lambda, Textract y Bedrock) y se verifica que todos operan en la cuenta de la empresa y en la región us-east-1. Resultado: los cinco servicios que reciben el documento o su contenido (SharePoint, S3,

Lambda, Textract y Bedrock) operan en la cuenta de AWS de la empresa y en la región us-east-1, ningún servicio externo lo recibe, frente a una meta de ninguno. Conviene precisar que este escenario no mide una mejora, el *bot* y el procedimiento manual también mantienen los documentos dentro de la empresa. Lo que E7 comprueba es que la solución preserva esa condición a pesar de incorporar un modelo de lenguaje, lo cual era una restricción de la sección 1.7.

4.4 COSTO Y ESFUERZO

Esta sección valida el escenario E8 en su dimensión económica y corresponde a la métrica de costo operacional. El costo de operar la solución no se reduce a la infraestructura, incluye las horas de las personas que la alimentan, revisan lo que no resuelve y la mantienen. Las horas y los dólares se presentan en filas separadas, sin convertir las primeras a dinero, porque no se dispone de un costo por hora que pueda citarse (Tabla 24).

Infraestructura. El costo de la solución se obtiene del registro de tokens y costo de cada factura procesada en la validación, llevado a las ~800 facturas mensuales y complementado con las tarifas de la Tabla 18 para los demás servicios. Conviene precisar una diferencia que juega en contra de la solución en este punto, la instancia del *bot* es compartida con el backend y la interfaz de EMER, de modo que su costo no desaparece al migrar los formatos. En términos de infraestructura, la solución agrega un costo variable, aunque bajo casi nulo, el ahorro, si existe, está principalmente en las horas.

Horas-persona. Para la solución se estiman tres componentes. El primero es el depósito de cada factura en la carpeta de su formato, que no cambia, porque la obtención quedó fuera del alcance. El segundo es la revisión de las facturas derivadas, la tasa medida en la sección 4.2, aplicada a 800 facturas, más las ~18 facturas mensuales compuestas por más de un documento, multiplicada por el tiempo que toma resolver cada una. El tercero es la mantención, que se estima a partir del esfuerzo medido en la sección 4.3.1 y de la frecuencia de cambios de formato observada con el *bot* (diez en tres meses), tomando ese esfuerzo como cota superior, porque ajustar un formato existente no debería costar más que incorporar uno nuevo.

Formatos restantes. El alcance comprometió medir el costo marginal de incorporar los formatos que siguen en el *bot* (sección 1.9). Se estima como el número de formatos por migrar, 25 (30 menos los 5 ya incorporados), multiplicado por las horas medidas en la sección 4.3.1, más el costo variable de sus facturas: 25 formatos × 2 a 4 horas = 50 a 100 horas, una sola vez, y unos USD 6 al mes por las 280 facturas de esos formatos, a USD 0,02 cada una. Se exponen los datos en la siguiente tabla.

Tabla 24: Costo y esfuerzo mensual de operación. Fuente: Elaboración propia a partir de datos de Evol.

Concepto	**Procedimiento manual**	***Bot***	**Solución**	**Tipo de dato**
Infraestructura (USD/mes)	-	120–130, fijo y compartido con EMER	~10	Empresa / medido
Infraestructura por factura (USD)	-	- (compartido, no atribuible)	0,02	Medido
Depósito, revisión y carga (h-p/mes)	60	18	6 a 10	Empresa / medido
Mantención y soporte (h-p/mes)	-	10 a 15	4 a 6	Empresa / medido
Esfuerzo total (h-p/mes)	60	28 a 33	10 a 16	Empresa / medido
Incorporar un formato (h)	-	~25	2 a 4	Empresa / medido
Migrar los formatos restantes (h, una vez)	-	-	50 a 100	Estimado

En síntesis, la meta del escenario E8 se cumple en horas y no en dólares. El esfuerzo mensual estimado de la solución, de 10 a 16 horas-persona, es entre un 73% y un 83% menor que el del procedimiento manual (60) y entre un 43% y un 70% menor que el del *bot* (28 a 33), incluyendo su mantención. En infraestructura, en cambio, la solución agrega un costo variable de unos USD 10 al mes sin eliminar el de la instancia compartida con EMER, por lo que el ahorro está en las horas de las personas. Incorporar los 25 formatos restantes demandaría entre 50 y 100 horas, una sola vez, y unos USD 6 mensuales adicionales. Estas cifras son estimaciones construidas a partir de lo medido, en particular las horas de mantención, cuyo supuesto se declara en los límites de la sección 4.6.

4.5 RESUMEN Y LÍMITES DE LA VALIDACIÓN

La Tabla 25 reúne el resultado de cada escenario frente a su meta y a las dos líneas base. Es la base con que el Capítulo 5 evalúa el cumplimiento de los objetivos.

Tabla 25: Resumen de la validación. Fuente: Elaboración propia.

Escenario	**Meta**	**Procedimiento manual**	***Bot***	**Solución**	**¿Cumple?**
E1: Precisión	≥ *bot*; ninguna carga sin validar	Sin medir	~55%	98% sobre procesadas; 89% sobre el total; 9 de 100 no procesadas; 39 campos incorrectos en 11 facturas cargadas, ninguno inventado	Sí: precisión ≥ *bot* en los cinco grupos; ninguna carga sin validar (por construcción)
E2: Formato nuevo	≤ 6 h; un servicio	-	~25 h; se despliega todo el proceso	2 a 4 h, un servicio	Sí
E3: Falla de EMER	Ninguna factura perdida ni sin estado	-	Sin reintentos; Requiere volver a depositar	Reintenta con espera creciente ~50 min; al agotarse, queda en ERROR con alerta; ninguna sin estado	Sí
E4: Aislamiento	Ninguna afectada por falla ajena	-	Aísla la falla de una factura, no la del proceso	Las facturas defectuosas van a revisión sin detener el lote; ninguna afectada	Sí

E5: Latencia	≤ 1h fuera del *peak*; ≤ 8 h en el *peak*	~2 semanas de ciclo	Hasta 24 h	Extracción de ~5 s por factura; fuera del *peak* ≤ ~16 min (derivado); 100 facturas cargadas en < 30 min	Sí, en ambas partes de la meta
E6: Estado consultable	100%	Sin registro	Solo errores, por correo	100% con estado consultable	Sí
E7: Confidencialidad	Ningún servicio externo	Dentro de la empresa	Dentro de la empresa	Todos los servicios en la cuenta de la empresa y en us-east-1	Sí
E8: *Peak* y costo	Sin intervención; costo < manual y ≤ *bot*	60 h-p/mes	28-33 h-p/mes y USD 120-130 compartido	Sin intervención manual; 10 a 16 h-p/mes y USD ~17/mes	Sí en horas; en USD agrega costo (el EC2 del *bot* es comparti-do)

Los resultados deben leerse junto con los límites de esta validación:

- Se realizó en el ambiente de calidad y no en producción, sin observar un *peak* mensual real.
- La muestra comprende 100 facturas de cuatro emisores en dos meses, no el volumen mensual completo ni los 30 formatos.
- La referencia de precisión proviene de EMER, cuyos datos pueden contener errores de carga, mitigados con la regla de la sección 4.1.3.
- Los montos se comparan con una tolerancia del 1%, fijada de forma experimental, y los consumos solo en su formato.
- El esfuerzo del *bot* por formato nuevo es una estimación del área de Desarrollo y Tecnología, y su comportamiento ante fallas se describe a partir de su funcionamiento, no de una prueba exhaustiva.
- Las horas-persona de la solución y su costo mensual son estimaciones construidas a partir de lo medido.

- La coordinación entre servicios se revisó durante el proyecto a pedido del área de Desarrollo y Tecnología (ADR-03); lo que se valida es la versión basada en colas, cuya visibilidad depende de que cada servicio registre su etapa, condición que verifica E6.
- La captura se autentica en SharePoint con usuario y contraseña, como el *bot*, la aplicación registrada que propone el diseño (sección 3.4) queda pendiente de la solicitud al área de Desarrollo y Tecnología, antes del paso a producción.

Ninguno de estos límites invalida la comparación, pero cada uno acota el alcance de sus conclusiones, que se retoman en el Capítulo 5.

CAPÍTULO 5: CONCLUSIONES

5.1 CONCLUSIONES GENERALES

Este trabajo partió de un proceso en que la carga de las facturas en EMER era el único paso manual de una cadena por lo demás automática. El procedimiento manual demandaba cerca de 60 horas-persona al mes, y la primera automatización, el *bot*, lo redujo a 28–33, pero falla en cerca del 30% de las facturas y su mantención consume más de un tercio de ese esfuerzo (Tabla 4). Los problemas de fondo eran tres: la fragilidad ante los cambios de formato, la imposibilidad de cambiar un extractor sin desplegar el proceso completo y la falta de visibilidad sobre lo que ocurre con cada factura. Para resolverlos se diseñó e implementó una arquitectura de cinco servicios en AWS, coordinados por colas, que extrae los datos con un modelo de lenguaje y los somete a una validación determinística antes de cargarlos. La validación en el ambiente de calidad, con 100 facturas reales, mostró que la solución cumple las metas de los ocho escenarios de calidad: supera al *bot* en precisión (98% frente a 91% sobre las facturas procesadas, y mayor en cuanto a facturas totales), extrae cada factura en unos 5 segundos, carga el conjunto de 100 facturas en menos de 30 minutos y reduce el esfuerzo mensual estimado a 10-16 horas-persona, frente a 28-33 del *bot*, no mejora, en cambio, en el costo de infraestructura ni en las facturas compuestas.

Más allá de ese resultado, el trabajo deja cuatro aprendizajes.

La complejidad de los microservicios se justifica por el cambio, no por el volumen. Cerca de 800 facturas al mes no exigen una arquitectura distribuida: un proceso único, bien diseñado, podría procesarlas en pocas horas. Lo que justifica dividirlo en servicios es la frecuencia con que cambia una parte (diez cambios de formato en tres meses) y el costo de que ese cambio afecte al resto. Separar la extracción permite ajustar un formato desplegando un solo servicio, con un esfuerzo de 2 a 4 horas al incorporar uno nuevo, frente a las cerca de 25 del *bot*, y separar las etapas mediante colas hace que la falla de una no detenga a las demás. El precio, anticipado en la Tabla 8, es real: más piezas que desplegar y vigilar, una política de errores que debe mantenerse común a todos los servicios y una depuración que exige seguir una factura por los registros de varias funciones. Para un equipo pequeño, ese precio solo se paga si la solución reemplaza al *bot* por completo, si ambos conviven indefinidamente, la empresa mantiene dos sistemas en lugar de uno.

El conocimiento del equipo que opera pesa tanto como el diseño. La coordinación del flujo se planteó al inicio como una orquestación con Step Functions, que reunía en un solo lugar la secuencia y los reintentos. Se reemplazó por colas a pedido del área de Desarrollo y Tecnología, que no había operado ese servicio (ADR-03). La decisión confirma lo que advierten Lewis y Fowler (2014): una arquitectura distribuida vale lo que vale la capacidad de quien la mantiene. El cambio no fue gratuito (la visibilidad pasó a depender de que cada servicio registre su etapa), pero la solución resultante usa solo tecnologías que el equipo ya conoce, lo que hace más probable que se mantenga después de esta memoria.

El modelo de lenguaje resuelve la variedad de formatos, pero la confianza la da la validación. Una sola instrucción, con añadidos breves por formato, reemplaza a los extractores escritos a mano para cada emisor. Sin embargo, un modelo puede devolver un valor plausible que no está en el documento, y nada en su respuesta lo distingue de uno correcto. Por eso la decisión más importante del diseño no fue el modelo, sino rodearlo de reglas determinísticas (estructura, cuadre de montos, conceptos del catálogo) y derivar a revisión todo lo que no las supera, en lugar de cargarlo. La medición muestra el alcance de esa protección: de las 91 facturas cargadas, 80 no tuvieron ningún error y 11 tuvieron algún campo incorrecto, 39 en total, ninguno de ellos un valor inventado. Son errores que superaron todas las reglas, porque eran valores coherentes con la estructura, el cuadre de montos y el catálogo, y solo se detectan al compararlos con una referencia, por eso se propone, como trabajo futuro, un seguimiento continuo de la precisión. Separar los cargos en reconocidos, excluidos y desconocidos llevó la detección de un concepto nuevo al momento previo a la carga, cuando con el *bot* ese mismo caso llegaba al cliente como una alerta.

Diseñar con metas explícitas hizo honesta la validación. Expresar cada atributo de calidad como un escenario con meta, antes de construir, convirtió el Capítulo 4 en una verificación y no en una demostración, cada resultado se compara con un valor comprometido de antemano, y lo que no lo alcanza queda a la vista.

Esa misma comparación muestra lo que la solución no mejora. Las facturas compuestas por dos archivos, que el *bot* combina cuando siguen su convención de nombre, la solución las deriva a revisión, porque su agregación quedó fuera del alcance, a cambio, ningún cobro se carga incompleto. En infraestructura, la solución agrega un costo variable, cercano a USD 17 al mes para las 800 facturas, a USD 0,02 cada una, sin eliminar el de la instancia del *bot*, que se comparte con EMER: el ahorro, cuando existe, está en las horas de las personas y no en la factura de la nube.

Estas conclusiones deben leerse con los límites de la sección 4.6. La validación se realizó en el ambiente de calidad, con una muestra de cuatro emisores y dos meses, y sin un *peak* mensual real; las horas-persona y el costo mensual son estimaciones construidas a partir de lo medido, y la solución depende de un proveedor de nube cuyo costo crece con el número y el largo de los documentos. Ninguno de estos límites cambia el sentido de los resultados, pero todos indican que la prueba definitiva será su operación en producción.

5.2 CUMPLIMIENTO DE OBJETIVOS

Esta sección revisa cada objetivo específico y, al final, el objetivo general, indicando en qué parte del trabajo se abordó, con qué evidencia y si se cumplió. Los resultados medidos se reúnen en la Tabla 25.

Objetivo específico 1. “Analizar el proceso actual de carga de facturas y la automatización existente, midiendo su desempeño e identificando sus limitaciones, para definir los requerimientos que debe cumplir la nueva solución.”

El Capítulo 1 describió el procedimiento manual y el *bot* (sección 1.3), los cuantificó en una línea base con fuente (Tabla 4) y organizó sus deficiencias en seis problemas del proceso (P1–P6) y siete limitaciones de la automatización (L1–L7), relacionados con los atributos de calidad y las métricas en la sección 1.6.3. El Capítulo 3 tradujo ese diagnóstico en once requerimientos funcionales (Tabla 12), ocho escenarios de calidad con meta (Tabla 13) y siete restricciones (Tabla 14). Un valor de la línea base no pudo medirse: la tasa de error de la carga manual, por lo que la precisión se comparó solo con el *bot*. Se declara logrado.

Objetivo específico 2. "Seleccionar los servicios en la nube para cada componente de la solución mediante criterios explícitos, considerando la infraestructura tecnológica existente en Evol Services."

La sección 3.2 definió cinco criterios (ajuste al atributo, operación administrada, costo según el uso, conocimiento del equipo y confidencialidad) y los aplicó a los trece componentes de la solución (Tabla 15), indicando en cada uno las alternativas, el servicio elegido y el criterio que decidió la elección. La infraestructura existente se consideró en dos planos: AWS como restricción (R1) y la experiencia del área de Desarrollo y Tecnología como criterio, que resultó decisivo al revisar la coordinación del flujo durante el proyecto (ADR-03). Se declara logrado.

Objetivo específico 3. "Diseñar la arquitectura de la solución, dividiendo el proceso en microservicios independientes y documentándola mediante diagramas y decisiones justificadas según los requerimientos definidos."

El Capítulo 3 dividió el proceso en cinco servicios (captura, extracción, validación, carga y estado) que se comunican mediante colas y se despliegan por separado (secciones 3.3 y 3.5). La arquitectura se documentó con los diagramas de contexto, y contenedores del modelo C4, el flujo entre colas, el ciclo de vida de una factura. La Tabla 16 muestra que cada uno de los ocho escenarios de calidad queda respaldado por al menos una decisión. Se declara logrado.

Objetivo específico 4. "Implementar los servicios que componen la arquitectura, incluyendo el seguimiento y reporte del estado de procesamiento de cada factura."

Los cinco servicios se implementaron y operan de extremo a extremo en el ambiente de calidad, incluidas la captura desde SharePoint y el traslado de cada documento a la carpeta que corresponde a su resultado (Tabla 19). La implementación cubre cinco emisores: los cuatro del conjunto de prueba y el que se incorporó para medir el esfuerzo de mantención (sección 4.3.1). El seguimiento del estado se ofrece por tres medios: la consulta por factura y por filtros, el resumen diario dirigido al equipo GCP y las carpetas de revisión y de error en SharePoint. En la validación, el 100% de las facturas tuvo su estado consultable (sección 4.3.4). Dos elementos quedaron pendientes: la agregación de documentos compuestos, fuera del alcance, y la autenticación mediante una aplicación registrada, reemplazada por ahora por usuario y contraseña. Se declara logrado.

Objetivo específico 5. "Validar la solución comparándola con el proceso manual y con la automatización preliminar, mediante métricas de precisión de extracción, tiempo de procesamiento, esfuerzo de mantención y costo operacional."

El Capítulo 4 validó la solución con 100 facturas reales de dos meses, procesadas también por el *bot* en un entorno separado, y comparó ambos resultados con la línea base del procedimiento manual. Las cuatro métricas del objetivo corresponden a los escenarios E1, E5, E2 y E8. La precisión de todos los campos fue de 98% en la solución y de 91% en el *bot*, sobre las facturas que procesó cada uno (89% y 66% sobre el total); el tiempo entre el depósito y la carga, de 20 s fuera del *peak* y de 35 min para las 100 facturas; incorporar un formato tomó 2 horas, frente a cerca de 25, y el esfuerzo mensual estimado fue de 10 a 16 horas-persona, frente a 60 y a 28-33. La comparación informó también lo que no mejora: las facturas de dos archivos, que el *bot* combina y la solución deriva a revisión, y el costo de infraestructura, que aumenta porque la instancia del *bot* se comparte con EMER. Se declara logrado.

Objetivo general. "Diseñar y validar una arquitectura de microservicios en la nube que automatice la extracción y carga de los datos de facturas en el sistema EMER de Evol Services, con el fin de reducir el esfuerzo manual y los errores del proceso actual." Los objetivos específicos primero a tercero cubren el diseño, y el cuarto y el quinto, su construcción y validación. La solución automatiza la extracción y la carga de los emisores implementados, y la validación muestra que reduce el esfuerzo manual de 28-33 a 10-16 horas-persona al mes respecto del *bot*, y de 60 respecto del procedimiento manual. En cuanto a los errores, la precisión sobre las facturas procesadas fue de 98%, frente a 91% del *bot*, y 11 de las 91 facturas cargadas tuvieron algún campo incorrecto, los conceptos desconocidos, que con el *bot* llegan al cliente como alerta, se detienen antes de la carga (Tabla 23). Se declara logrado el objetivo general, dentro de los límites expuestos en la sección 4.6.

5.3 TRABAJO FUTURO

Las líneas de trabajo que se desprenden de esta memoria se ordenan según su valor para la operación de la empresa.

1. **Paso a producción**. Operar la solución durante un *peak* mensual completo, en paralelo con el *bot* en los emisores ya migrados. Antes, conviene reemplazar la autenticación con usuario y contraseña por la aplicación registrada (ADR-07) y trasladar las credenciales a un servicio de gestión de secretos (sección 3.6.1).
2. **Migración de los formatos restantes**. Incorporar los 25 formatos que siguen en el *bot*, con un esfuerzo estimado de 50 a 100 horas (sección 4.4), hasta retirarlo.
3. **Agregación de documentos compuestos**. Implementar el mecanismo diseñado en la sección 3.7, que permitiría cargar de una vez las facturas de dos archivos y las acompañadas de una proforma, hoy derivadas a revisión. Es el único caso en que la solución hace menos que el *bot*.

4. **Catálogo de conceptos**. Convertir la marca de obligatorio del catálogo de EMER en una categoría (obligatorio, opcional o excluido) para mantener en un solo lugar lo que hoy reparten EMER y la tabla de configuración (sección 3.5.3), evaluar un umbral por monto para que los cargos desconocidos menores no detengan la factura, y reducir la tolerancia de comparación de los montos a medida que se conozca el comportamiento de cada emisor (sección 4.2.1).
5. **Estado visible en EMER**. Consultar el estado de cada factura desde la interfaz de EMER, aprovechando que la consulta ya usa su mismo mecanismo de autenticación (sección 3.5.5).
6. **Nuevas entradas**. Incorporar los documentos escaneados y los protegidos con contraseña.
7. **Seguimiento continuo de la precisión**. Comparar cada mes una muestra de facturas cargadas con su documento, con el mismo método de la sección 4.2, para detectar cuándo un emisor cambia su formato o el modelo empeora en algún campo, antes de que el problema se refleje en los reportes enviados a los clientes.

5.4 PALABRAS FINALES

Lo más difícil de este trabajo no fue escribir los servicios, sino decidir cómo debía ser la arquitectura. Un mismo problema admite muchas soluciones razonables, y elegir la que corresponde exigió mirar más allá de las facturas: qué tecnologías usa la empresa, qué sabe operar su equipo y cómo trabaja. Nada de eso aparece en la descripción del problema, lo entrega el contexto al que la solución tiene que llegar.

Trabajar dentro de Evol me dejó otra lección: siempre puede aparecer un caso borde que nadie había previsto. Adaptarse y reaccionar a tiempo resultó tan importante como el diseño inicial, y por eso intenté que la solución también fuera fácil de ajustar cuando el próximo caso aparezca.

Uso de inteligencia artificial: Durante el trabajo se utilizaron Claude y Claude Code, de Anthropic, como herramientas de apoyo. Claude Code se empleó principalmente para escribir código en la implementación de la solución, entre el diseño descrito en el Capítulo 3 y la validación del Capítulo 4, herramienta que la empresa ya utiliza en su trabajo de desarrollo. Claude se empleó, además, para encontrar fuentes bibliográficas con mayor rapidez, las que fueron posteriormente revisadas por el autor. Las decisiones de diseño, el análisis de los resultados y las conclusiones son responsabilidad del autor.

Agradezco al coordinador del área de Desarrollo y Tecnología, mi jefe, que fue un pilar en las decisiones técnicas, y al coordinador del equipo GCP, referente en la lógica del negocio y del proceso. Sus apoyo se complementó, y su disponibilidad para responder cada consulta fue lo que permitió avanzar.

Me llevo, en primer lugar, la satisfacción de haber puesto a prueba lo aprendido en la carrera en un ambiente real, lo que da seguridad para el trabajo que viene. Y me llevo también la convicción de que no hay proceso de negocio, por ajeno que parezca a la informática, en el que ella no pueda aportar.

REFERENCIAS BIBLIOGRÁFICAS

Ministerio de Economía, Fomento y Reconstrucción. (5 de Febrero de 2007). *Biblioteca del Congreso Nacional de Chile.* Obtenido de Decreto con Fuerza de Ley N° 4/20.018. Ley General de Servicios Eléctricos: https://www.tdlc.cl/normativa---otras-le/dfl-4-20-018-ley-general-de-servicios-electricos/

Comisión de Minería y Energía del Senado. (Septiembre de 2019). *Asesoría Técnica Parlamentaria.* Obtenido de Traspaso de clientes regulados a libres: https://obtienearchivo.bcn.cl/obtienearchivo?id=repositorio%2F10221%2F27778%2F1%2FBCN___Traspaso_de_clientes_regulados_a_libres_sept2019.pdf

Ministerio de Energía. (9 de Diciembre de 2024). *Biblioteca del Congreso Nacional de Chile.* Obtenido de Resolución Exenta N° 58: Rebaja límite de capacidad instalada para optar a ser cliente libre, de acuerdo a lo dispuesto por el artículo 147°, literal d) de la Ley General de Servicios Eléctricos: https://www.bcn.cl/leychile/navegar?idNorma=1209158

Bass, L., Kazman, R., & Clements, P. (2021). *Software Architecture in Practice, 4th Edition.* Obtenido de Software architecture in practice (4.ª ed.).

ISO/IEC. (2023). *ISO/IEC 25010:2023 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Product quality model.*

Cervantes, H., & Kazman, R. (2016). *Designing Software Architectures: A Practical Approach.*

Brown, S. (s.f.). *C4 model*. Obtenido de The C4 model for visualising software architecture: https://c4model.com/

Richards, M., & Ford, N. (2020). *Fundamentals of Software Architecture.*

Lewis, J., & Fowler, M. (25 de Marzo de 2014). *martinfowler*. Obtenido de Microservices a definition of this new architectural term: https://martinfowler.com/articles/microservices.html

Newman, S. (2021). *Building Microservices, 2nd Edition.*

Mell, P., & Grance, T. (2001). *The NIST Definition of Cloud Computing.* National Institute of Standards and Technology.

Golec, M., Walia, G. K., Kumar, M., Cuadrado, F., Gil, S. S., & Uhlig, S. (2024). Cold Start Latency in Serverless Computing: A Systematic Review, Taxonomy, and Future Directions . *ACM Computing Survey*, 1-36.

Abiteboul, S. (1997). Querying semi-structured data. *Database Theory*, 1-18.

Yashwant, S., Dubey, A., Paikray, P., & Thulsiram, G. (22 de Octubre de 2025). *Invoice Information Extraction: Methods and Performance Evaluation.* Obtenido de arxiv: https://arxiv.org/abs/2510.15727

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2019). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery*, 1192–1200.

Wang, W., Hu, H., Zhang, Z., Li, Z., Shao, H., & Dahlmeier, D. (15 de Octubre de 2025). *Document Intelligence in the Era of Large Language Models: A Survey.* Obtenido de arxiv: https://arxiv.org/abs/2510.13366

Perot, V., Kang, K., Luisier, F., Su, G., Sun, X., Boppana, R. S., . . . Hua, N. (2024 de Junio de 2024). *LMDX: Language Model-based Document Information Extraction and Localization.* Obtenido de arxiv: https://arxiv.org/abs/2309.10952

Khanchandani, K., Thakur, A., Shetty, A., Reddy, C., & Behera, R. (8 de Enero de 2026). *Automated Invoice Data Extraction: Using LLM and OCR.* Obtenido de arxiv: https://arxiv.org/abs/2511.05547

Microsoft. (2017). *driveItem: delta*. Obtenido de Learn Microsoft Graph: https://learn.microsoft.com/en-us/graph/api/driveitem-delta?view=graph-rest-1.0&tabs=http

Microsoft. (2024a). *Set up notifications for changes in resource data*. Obtenido de Learn Microsoft Graph: https://learn.microsoft.com/en-us/graph/change-notifications-overview

Microsoft. (2024b). *Overview of Selected permissions in OneDrive and SharePoint*. Obtenido de Learn Microsoft Graph: https://learn.microsoft.com/en-us/graph/permissions-selected-overview?tabs=http

Microsoft. (2025). *Microsoft identity platform and OAuth 2.0 Resource Owner Password Credentials*. Obtenido de Learn Microsoft Entra: https://learn.microsoft.com/en-us/entra/identity-platform/v2-oauth-ropc

Amazon Web Services. (2026a). *Amazon SQS standard queues*. Obtenido de Amazon Docs: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/standard-queues.html

Amazon Web Services. (2026b). *Lambda quotas*. Obtenido de AWS Lambdas Developer Guide: https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html

Amazon Web Services. (2026c). *Guidance for Accelerated Intelligent Document Processing on AWS*. Obtenido de Amazon Docs: https://docs.aws.amazon.com/solutions/accelerated-intelligent-document-processing-on-aws/

Amazon Web Services. (2026d). *Analyzing invoices and Receipts*. Obtenido de Amazon Docs: https://docs.aws.amazon.com/textract/latest/dg/invoices-receipts.html

Amazon Web Services. (2026e). *Data protection*. Obtenido de Amazon Bedrock User Guide: https://docs.aws.amazon.com/bedrock/latest/userguide/data-protection.html

Amazon Web Services. (2026f). *Choosing workflow type in Step Functions*. Obtenido de AWS Step Functions Developer Guide: https://docs.aws.amazon.com/step-functions/latest/dg/choosing-workflow-type.html

Amazon Web Services. (2026g). *Constraints in Amazon DynamoDB*. Obtenido de Amazon DynamoDB User Guide: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Constraints.html

Amazon Web Services. (2026h). *Configuring reserved concurrency for a function*. Obtenido de AWS Lambda Developer Guide: https://docs.aws.amazon.com/lambda/latest/dg/configuration-concurrency.html

Mays, J. A., & Mathias, P. C. (2019). Measuring the rate of manual transcription error in outpatient point-of-care testing. *J Am Med Inform Assoc*, 269-272.

Panko, R. (2008). Thinking is Bad: Implications of Human Error Research for Spreadsheet Research and Practice. *arxiv*.