



**Universidad
Católica de
Santa María**

Facultad de Ciencias e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería de Sistemas

**Implementación de una solución de inteligencia de negocios para optimizar
la generación de reportes de seguimiento de microcréditos en una empresa
financiera en la ciudad de Arequipa**

Tesis presentada por:

Urrutia Quequezana, Gonzalo

ORCID: 0000-0001-5365-8824

Para optar el Título Profesional de Ingeniero de Sistemas

Asesora:

Dra. Rosas Paredes, Karina

ORCID: 0000-0003-4650-7432

Arequipa – Perú

2026

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

INGENIERIA DE SISTEMAS

TITULACIÓN CON TESIS

DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 21 de Abril del 2026

Dictamen: 013530-C-EPIS-2026

Visto el borrador del expediente 013530, presentado por:

2019205381 - URRUTIA QUEQUEZANA GONZALO

Titulado:

**IMPLEMENTACIÓN DE UNA SOLUCIÓN DE INTELIGENCIA DE NEGOCIOS PARA OPTIMIZAR LA
GENERACIÓN DE REPORTES DE SEGUIMIENTO DE MICROCRÉDITOS EN UNA EMPRESA
FINANCIERA EN LA CIUDAD DE AREQUIPA**

Nuestro dictamen es:

APROBADO

Título Profesional/Título de Segunda Especialidad/Grado Académico a optar:

INGENIERO DE SISTEMAS

**29643112 - GUEVARA PUENTE DE LA VEGA KARIM
DICTAMINADOR**



**29591972 - CALDERON RUIZ GUILLERMO ENRIQUE
DICTAMINADOR**



**71132586 - ANGULO OSORIO JAVIER FERNANDO
DICTAMINADOR**



Implementación de una solución de inteligencia de negocios para optimizar la generación de reportes de seguimiento de microcréditos en una empresa financiera en la ciudad de Arequipa

INFORME DE ORIGINALIDAD



FUENTES PRIMARIAS

1	Submitted to Universidad Católica de Santa María	1%
	Trabajo del estudiante	
2	hdl.handle.net	<1%
	Fuente de Internet	
3	repositorio.ucsm.edu.pe	<1%
	Fuente de Internet	
4	tesis.ucsm.edu.pe	<1%
	Fuente de Internet	
5	docplayer.es	<1%
	Fuente de Internet	
6	Submitted to CSU, San Jose State University	<1%
	Trabajo del estudiante	
7	repositorio.unp.edu.pe	<1%
	Fuente de Internet	
8	Submitted to Universidad San Ignacio de Loyola	<1%
	Trabajo del estudiante	
9	Ribeiro, Margarida Isabel Cerqueira. "Stock Management in Automated Pharmaceutical Warehouses: Improved Detection and	<1%

Dedicatoria

Dedico esta tesis a mis padres, Mario y Rocío, por ser mi motivación, mi ejemplo y por acompañarme con su apoyo incondicional en cada etapa de mi vida.

A mis abuelas: Irma, por su amor infinito y por cuidarme desde el cielo; y Betty, cuyo amor siempre ha sido y será un refugio para mí.

A mis hermanos, Bruno y Christian, por su cariño y por estar siempre a mi lado.

Finalmente a Dios, por ser la base y guía de mi vida.



Agradecimientos

Agradezco a Dios por ser mi fortaleza y mi guía en cada etapa de mi vida.

A mis padres, por acompañarme siempre y por apoyarme en mi crecimiento personal y profesional.

Finalmente, agradezco a mis docentes, quienes no solo compartieron su conocimiento académico, sino que también me brindaron valiosos consejos humanos que marcaron esta etapa.



RESUMEN

El presente proyecto implementa una solución de inteligencia de negocios para optimizar la generación de reportes de seguimiento de microcréditos en Holding Waki SAC, una empresa financiera ubicada en Arequipa (Holding Waki S.A.C., 2024). La problemática identificada consistía en la elaboración manual de reportes operativos y gerenciales mediante hojas de cálculo, proceso que consumía entre 1 y 1.5 días hábiles, presentaba tasas de error del 25.0 % al 33.3 %, generaba retrasos promedio de 2.3 a 3.5 días en la entrega de información, y demandaba 51 horas mensuales adicionales de retrabajo por parte del asistente del área, equivalente a S/. 5 737.50 anuales. (Administración de Holding Waki S.A.C., 2025).

La solución desarrollada integra un proceso ETL automatizado implementado con funciones AWS Lambda que extraen datos desde Apache Cassandra, un Data Warehouse en Amazon Redshift Serverless basado en un modelo dimensional con esquema estrella, y dashboards interactivos en Microsoft Power BI. Se adoptó la metodología de Ralph Kimball para el diseño del Data Warehouse, implementando un datamart de microcréditos con granularidad a nivel de cuota y cinco dimensiones: cliente, crédito, tiempo, producto y estado de pago.

En cuanto a infraestructura, se emplea una arquitectura serverless utilizando diferentes servicios en AWS, EventBridge coordina los procesos automáticos y la conexión de Power BI con el Data Warehouse varía según el entorno, utilizando túneles SSM para el ambiente de desarrollo y *On-premises Data Gateway* para el ambiente de producción, manteniendo todo dentro de una VPC privada.

Los resultados demuestran mejoras significativas en los cuatro indicadores clave: reducción del 99.98 % en el tiempo de elaboración de reportes (de 480 y 720 minutos a aproximadamente 7 segundos de carga del dashboard), eliminación completa de retrasos en la entrega de información, eliminación total de errores en los datos y supresión del retrabajo.

Palabras clave: Inteligencia de negocios, Data Warehouse, Microcréditos.

ABSTRACT

This project implements a business intelligence solution to optimize report generation for microcredit monitoring at Holding Waki SAC, a financial company based in Arequipa (Holding Waki S.A.C., 2024). The identified problem was the manual preparation of operational and managerial reports using spreadsheets, a process that took between 1 and 1.5 business days, had error rates ranging from 25.0% to 33.3%, caused average delivery delays of 2.3 to 3.5 days, and demanded an additional 51 monthly hours of rework from the area's assistant, equivalent to S/. 5 737.50 per year (Administración de Holding Waki S.A.C., 2025).

The developed solution integrates an automated ETL process built with AWS Lambda functions that extracts data from Apache Cassandra, a Data Warehouse on Amazon Redshift Serverless based on a dimensional model with a star schema, and interactive dashboards in Microsoft Power BI. Ralph Kimball's methodology was adopted for the Data Warehouse design, implementing a microcredit datamart with installment-level granularity and five dimensions: customer, credit, time, product, and payment status.

Regarding infrastructure, the system runs on a serverless architecture across multiple AWS services, EventBridge handles process automation, and the connection between Power BI and the Data Warehouse differs by environment, using SSM tunnels for development and *On-premises Data Gateway* for production, keeping everything within a private VPC.

The results show significant improvements across all four key indicators: a 99.98% reduction in report preparation time (from 480 and 720 minutes to approximately 7 seconds of dashboard load), full elimination of information delivery delays, full elimination of data errors, and suppression of manual rework.

Keywords: Business Intelligence, Data Warehouse, Microcredits.

ÍNDICE

DEDICATORIA

AGRADECIMIENTOS

RESUMEN

ABSTRACT

INTRODUCCIÓN..... 1

CAPÍTULO I 2

1 DESCRIPCIÓN DEL PROYECTO 3

1.1. Objetivos 3

1.1.1. General 3

1.1.2. Específicos 3

1.2. Alcances y limitaciones 3

1.3. Situación actual y línea base 4

1.3.1. Descripción de la situación actual 4

1.3.2. Definición de métricas para línea base 4

1.3.2.1 Tiempo de elaboración de reportes 4

1.3.2.2 Oportunidad de la información 6

1.3.2.3 Exactitud y consistencia de la información 7

1.3.2.4 Retrabajo sobre el proceso 7

1.4. Fundamentos Teóricos 9

1.4.1. Antecedentes del proyecto 9

1.4.2. Bases Teóricas del proyecto 11

1.4.2.1 Inteligencia de Negocios 11

1.4.2.2 Metodología de Ralph Kimball 11

1.4.2.3 Data Warehouse 12

1.4.2.4 Tabla de Hechos 12

1.4.2.5 Tablas Dimensionales 12

1.4.2.6 ETL 12

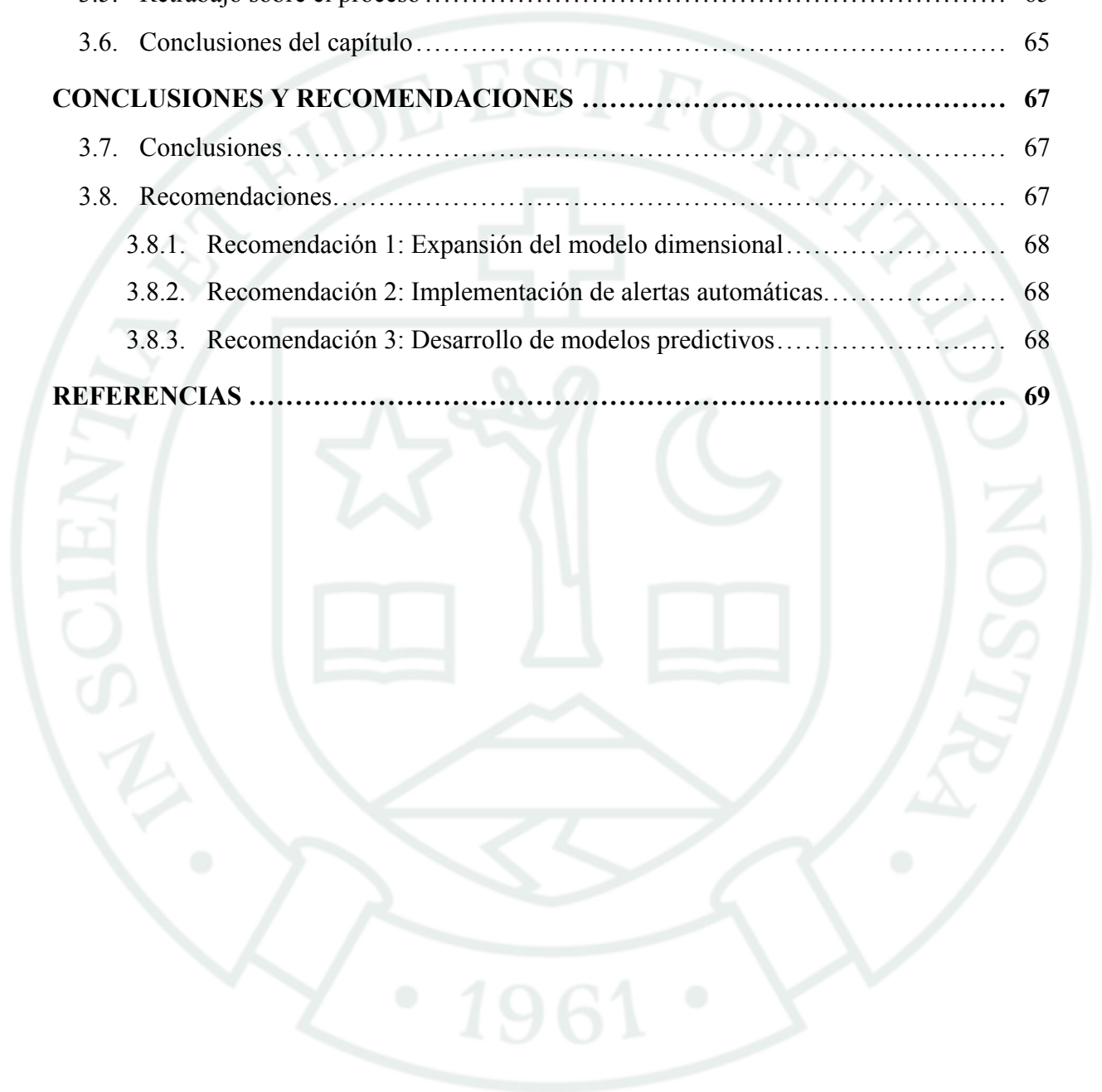
1.4.2.7 Amazon Redshift 12

1.4.2.8 Amazon EventBridge 12

1.4.2.9	AWS Lambda	13
1.4.2.10	Microsoft PowerBI	13
1.4.3.	Técnicas y Herramientas	13
1.4.3.1	Comparación de metodologías para el diseño del Data Warehouse	13
1.4.3.2	Comparación de herramientas para el desarrollo del ETL	14
1.4.3.3	Comparación de tecnologías para el almacenamiento analítico	15
1.4.3.4	Herramientas de visualización	15
CAPÍTULO II		16
2	DESARROLLO DE LA SOLUCIÓN DE INTELIGENCIA DE NEGOCIOS	17
2.1.	Visión general del ciclo de vida y metodología	17
2.1.1.	Justificación del ciclo de vida seleccionado	17
2.1.2.	Adaptación de la metodología al proyecto	17
2.2.	Análisis funcional y de datos	18
2.2.1.	Requerimientos de información	18
2.2.2.	Fuentes de datos y reglas de negocio	19
2.3.	Diseño de la arquitectura de la solución	21
2.3.1.	Arquitectura general en la nube	21
2.3.2.	Componentes principales	22
2.3.3.	Flujos de datos y orquestación	24
2.3.3.1	Orquestación con Amazon EventBridge Scheduler	25
2.3.3.2	Puntos de control y monitoreo	26
2.4.	Diseño del modelo dimensional	26
2.4.1.	Definición de la granularidad	26
2.4.2.	Dimensiones del <i>datamart</i> de microcréditos	27
2.4.2.1	Atributos de la Dimensión Cliente	28
2.4.2.2	Atributos de la Dimensión Crédito	29
2.4.2.3	Atributos de la Dimensión Estado de Pago	29
2.4.2.4	Atributos de la Dimensión Producto	30
2.4.2.5	Atributos de la Dimensión Tiempo	30
2.4.3.	Flujo de datos del proceso ETL	31
2.4.4.	Tabla de hechos	32

2.4.5.	Diagrama del modelo dimensional.....	32
2.5.	Diseño e implementación del proceso ETL	34
2.5.1.	Extracción desde Cassandra	34
2.5.2.	Transformaciones aplicadas	34
2.5.2.1	Conversión de tipos de datos	34
2.5.2.2	Limpieza y estandarización	35
2.5.2.3	Integración de datos (merges)	36
2.5.2.4	Aplicación de reglas de negocio	36
2.5.2.5	Cálculo de campos derivados	37
2.5.3.	Carga en Amazon Redshift.....	37
2.5.3.1	Estrategia de carga completa (Full Load)	38
2.5.3.2	Estrategia de carga incremental (Upsert)	39
2.5.4.	Rendimiento del proceso ETL.....	40
2.5.5.	Aseguramiento de calidad del ETL.....	43
2.6.	Diseño e implementación de dashboards en Power BI	45
2.6.1.	Conexión con el Data Warehouse.....	45
2.6.1.1	Arquitectura de conexión híbrida	45
2.6.1.2	Flujo de trabajo en entorno de desarrollo	46
2.6.1.3	Flujo de trabajo en entorno de producción	47
2.6.1.4	Consideraciones de seguridad y optimización de costos	48
2.6.2.	Estructura y diseño de los dashboards.....	49
2.6.2.1	Página 1: Reporte Gerencial - Empresas	50
2.6.2.2	Página 2: Reporte Gerencial - Empresas Detalle	51
2.6.2.3	Página 3: Reporte Operativo - Cobranza	52
2.6.2.4	Página 4: Reporte Operativo - Alertas y Riesgo	53
2.6.3.	Validación con el usuario final	54
2.7.	Resumen del capítulo.....	54
CAPÍTULO III		56
3 RESULTADOS		57
3.1.	Introducción.....	57
3.2.	Tiempo de elaboración de reportes.....	57

3.3. Oportunidad de la información	59
3.3.1. Evidencia objetiva de la actualización automática.....	59
3.4. Exactitud de la información.....	61
3.4.1. Reconciliación de volúmenes entre Cassandra y Redshift	62
3.5. Retrabajo sobre el proceso	65
3.6. Conclusiones del capítulo.....	65
CONCLUSIONES Y RECOMENDACIONES	67
3.7. Conclusiones	67
3.8. Recomendaciones.....	67
3.8.1. Recomendación 1: Expansión del modelo dimensional.....	68
3.8.2. Recomendación 2: Implementación de alertas automáticas.....	68
3.8.3. Recomendación 3: Desarrollo de modelos predictivos.....	68
REFERENCIAS	69



ÍNDICE DE TABLAS

Tabla 1	Análisis de tiempos para la elaboración del Reporte Operativo	5
Tabla 2	Análisis de tiempos para la elaboración del Reporte Gerencial	5
Tabla 3	Análisis de oportunidad en la entrega de reportes (últimos 3 meses)	6
Tabla 4	Análisis de exactitud de la información en reportes (últimos 3 meses)	7
Tabla 5	Análisis de retrabajo sobre el proceso de elaboración de reportes	8
Tabla 6	Costo económico del retrabajo (línea base).....	8
Tabla 7	Comparación de metodologías de diseño de Data Warehouse	13
Tabla 8	Comparativa de herramientas ETL en la nube	14
Tabla 9	Comparativa de tecnologías de almacenamiento analítico	15
Tabla 10	Comparación de herramientas BI.....	15
Tabla 11	Programación de ejecución de procesos ETL.....	25
Tabla 12	Atributos de la Dimensión Cliente.....	28
Tabla 13	Atributos de la Dimensión Crédito	29
Tabla 14	Atributos de la Dimensión Estado de Pago	29
Tabla 15	Atributos de la Dimensión Producto	30
Tabla 16	Atributos de la Dimensión Tiempo	30
Tabla 17	Duración de ejecución de las funciones Lambda del ETL	43
Tabla 18	Cobertura de pruebas unitarias por componente del ETL	44
Tabla 19	Estructura del sistema de dashboards	49
Tabla 20	Comparación del tiempo de elaboración de reportes antes y después de la implementación.....	58
Tabla 21	Comparación de la oportunidad de la información antes y después de la implementación.....	59
Tabla 22	Comparación de la exactitud de la información antes y después de la implementación.....	61

Tabla 23	Reconciliación de volúmenes entre Cassandra y Redshift	64
Tabla 24	Comparación del retrabajo sobre el proceso antes y después de la implementación	65
Tabla A.1	Configuración de Redshift Serverless	77
Tabla A.2	Resumen de tablas del datamart	79
Tabla A.3	Diccionario de datos - dim_cliente (demografía)	83
Tabla A.4	Diccionario de datos - dim_cliente (información laboral)	84
Tabla A.5	Diccionario de datos - dim_cliente (identificación)	84
Tabla A.6	Diccionario de datos - dim_credito (identificación y montos)	84
Tabla A.7	Diccionario de datos - dim_credito (tasas y costos)	85
Tabla A.8	Diccionario de datos - dim_credito (fechas y estado)	85
Tabla A.9	Diccionario de datos - dim_estado_pago	85
Tabla A.10	Diccionario de datos - dim_producto	86
Tabla A.11	Diccionario de datos - dim_tiempo	86
Tabla A.12	Diccionario de datos - fact_cuota_microcredito (medidas)	87
Tabla A.13	Diccionario de datos - fact_cuota_microcredito (claves)	87
Tabla A.14	Diccionario de datos - fact_cuota_microcredito (atributos generados)	88
Tabla A.15	Valores de dim_producto	88
Tabla A.16	Valores de dim_estado_pago	89
Tabla A.17	Funciones Lambda y fuentes de datos	89
Tabla A.18	Configuración técnica de funciones Lambda	90
Tabla A.19	Variables de entorno de las funciones Lambda	90
Tabla A.20	Componentes de la arquitectura de conexión	91
Tabla A.21	Parámetros de conexión Power BI Desktop	94
Tabla A.22	Configuración del On-premises Data Gateway	95
Tabla A.23	Regla de Security Group para Redshift	96
Tabla A.24	Errores comunes y soluciones	98



ÍNDICE DE FIGURAS

Figura 1	Keyspaces de los microservicios de Cassandra	19
Figura 2	Arquitectura del ETL	22
Figura 3	Funciones Lambda del ETL en AWS Console.....	23
Figura 4	Repositorio ECR con imagen Docker del ETL	24
Figura 5	Credenciales almacenadas en AWS Secrets Manager	24
Figura 6	Schedules del ETL en Amazon EventBridge Scheduler	25
Figura 7	Flujo de datos del proceso ETL	31
Figura 8	Modelo dimensional del <i>datamart</i> de microcréditos	33
Figura 9	Tiempo de ejecución de etl-dim-estado-pago en CloudWatch	40
Figura 10	Tiempo de ejecución de etl-dim-tiempo en CloudWatch	41
Figura 11	Tiempo de ejecución de etl-dim-producto en CloudWatch	41
Figura 12	Tiempo de ejecución de etl-dim-cliente en CloudWatch	41
Figura 13	Tiempo de ejecución de etl-dim-credito en CloudWatch	42
Figura 14	Tiempo de ejecución de etl-fact-cuota en CloudWatch.....	42
Figura 15	Resultado de ejecución de las 37 pruebas unitarias del ETL.....	44
Figura 16	Arquitectura híbrida de conexión entre Power BI y Amazon Redshift Serverless	46
Figura 17	Dashboard Gerencial - Vista General de Empresas	50
Figura 18	Dashboard Gerencial - Detalle por Empresa	51
Figura 19	Dashboard Operativo - Cobranza.....	52
Figura 20	Dashboard Operativo - Alertas y Riesgo	53
Figura 21	Historial de ejecuciones diarias de etl-fact-cuota en CloudWatch Logs	60
Figura 22	Conteo de microcréditos en Cassandra	62
Figura 23	Conteo de microcréditos en Redshift (dw_microcreditos.dim_credito)	62
Figura 24	Conteo de cuotas en Cassandra (payment_schedule)	62

Figura 25 Conteo de cuotas en Redshift (dw_microcreditos.fact_cuota)	63
Figura 26 Conteo de usuarios en Cassandra (user_management_ms.user)	63
Figura 27 Conteo de clientes en Redshift (dw_microcreditos.dim_cliente).....	63
Figura A.1 Configuración del Workgroup en Amazon Redshift Serverless.....	78
Figura A.2 Configuración del Namespace en Amazon Redshift Serverless	78
Figura A.3 Consulta de datos en Redshift Query Editor	79
Figura A.4 Consola de AWS EC2 mostrando la instancia PBI-Gateway-Windows en estado Running.....	92
Figura A.5 Terminal mostrando el túnel SSM activo con mensaje “Waiting for connections”	93
Figura A.6 Archivo hosts de Windows mostrando la entrada DNS personalizada	93
Figura A.7 Diálogo de conexión de Power BI Desktop con los parámetros configurados	94
Figura A.8 Tablas del esquema dw_microcreditos en Power BI	95
Figura A.9 Consola de Power BI Service mostrando el Gateway en estado Online	96
Figura A.10 AWS Security Group mostrando la regla de entrada configurada para Redshift	96
Figura A.11 Configuración de conexión en Power BI Service con el Gateway	97
Figura A.12 Configuración de mapeo de Gateway para publicación de dashboard	108
Figura B.1 Acta de aprobación de KPIs firmada por la administradora — Página 1	111
Figura B.2 Acta de aprobación de KPIs firmada por la administradora — Página 2	112
Figura C.1 Acta de aprobación del diseño visual firmada por la administradora — Página 1.....	113
Figura C.2 Acta de aprobación del diseño visual firmada por la administradora — Página 2.....	114
Figura C.3 Acta de aprobación del diseño visual firmada por la administradora — Página 3.....	115
Figura C.4 Acta de aprobación del diseño visual firmada por la administradora — Página 4.....	116

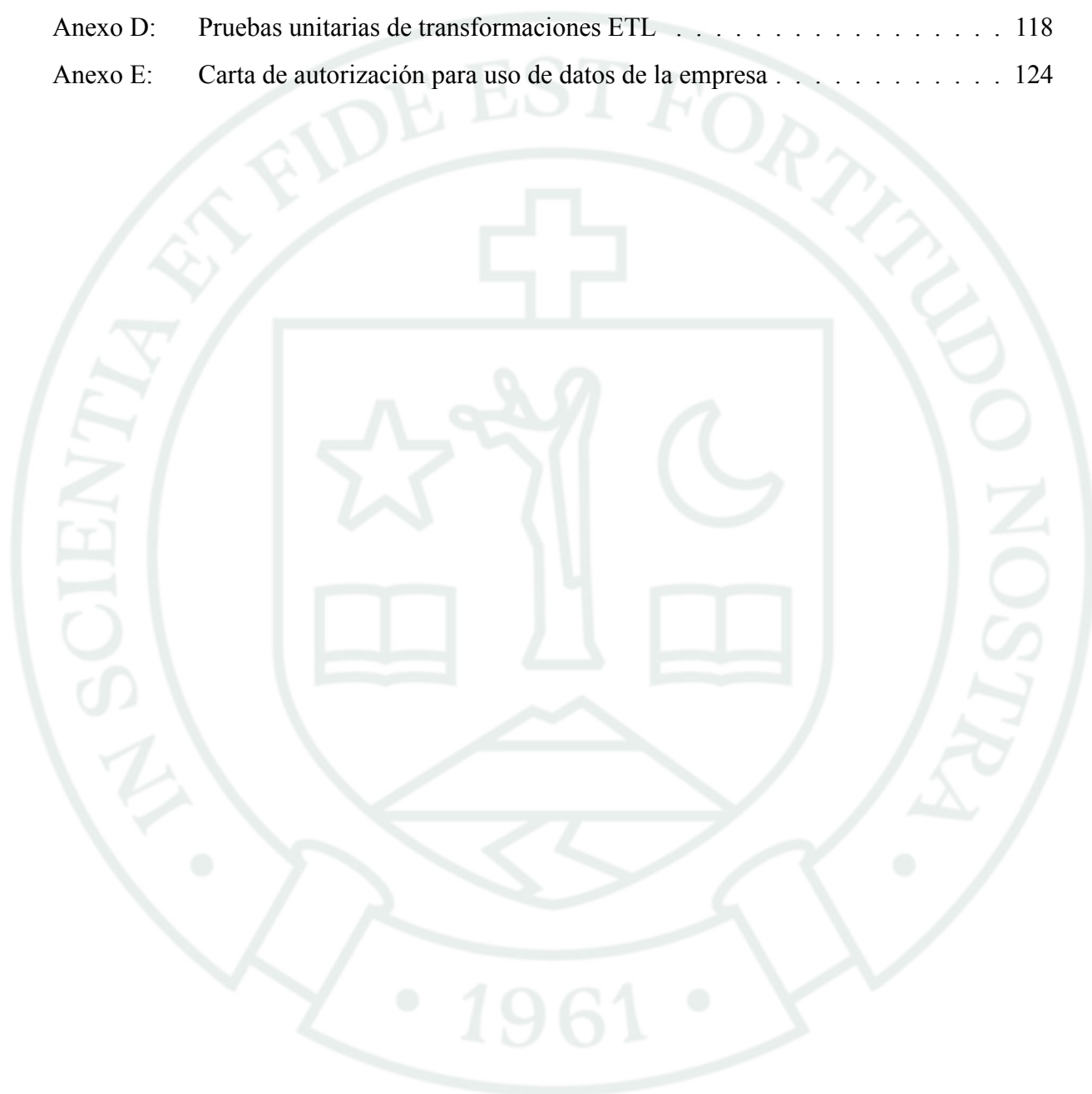
Figura C.5 Acta de aprobación del diseño visual firmada por la administradora — Pá-
gina 5..... 117

Figura E.1 Carta de autorización para uso de datos de la empresa 124



ÍNDICE DE ANEXOS

Anexo A:	Documentación técnica de la solución	72
Anexo B:	Acta de aprobación de KPIs	111
Anexo C:	Acta de aprobación del diseño visual de dashboards	113
Anexo D:	Pruebas unitarias de transformaciones ETL	118
Anexo E:	Carta de autorización para uso de datos de la empresa	124



INTRODUCCIÓN

En los últimos años, la transformación digital ha cambiado la forma en que las organizaciones gestionan sus datos. El sector financiero es uno de los más involucrados en esta transformación; la necesidad de tomar decisiones rápidas y fundamentadas ha impulsado la adopción de herramientas de análisis de datos que permiten convertir grandes volúmenes de información operativa en indicadores útiles para la gestión (Kanchepeu, 2023). Sin embargo, en el segmento de las microfinanzas (*Fintech*), muchas organizaciones aún operan con procesos manuales de elaboración de reportes, como el uso de hojas de cálculo, lo que genera demoras, errores y una dependencia excesiva del trabajo individual de quienes los preparan.

Esta problemática no es exclusiva de una empresa en particular. Mora (2020) analiza la influencia de la inteligencia de negocios en los procesos de toma de decisiones dentro de instituciones financieras en América Latina, concluyendo que la dependencia de reportes manuales limita significativamente la capacidad de respuesta de estas organizaciones ante cambios en su entorno operativo. En el contexto peruano, experiencias como la de Retuerto, Tuero, y Andrade-Arenas (2023) confirman que la implementación de soluciones BI en entidades del sector financiero permite mejorar el acceso a información consolidada y reducir los tiempos de generación de reportes, con impactos positivos en la toma de decisiones estratégicas.

Holding Waki SAC es una empresa financiera ubicada en Arequipa enfocada al otorgamiento de microcréditos a trabajadores de empresas (Holding Waki S.A.C., 2024). Con una cartera que ha crecido de forma sostenida en los últimos años, la administradora de la empresa elabora manualmente dos reportes periódicos: uno operativo para el área de créditos y otro de carácter gerencial para la junta de socios (Administración de Holding Waki S.A.C., 2025). Este proceso manual está sujeto a errores de transcripción y cálculo, además de generar retrasos en la entrega de información a quienes la necesitan para tomar decisiones.

Por este motivo, el presente proyecto tiene como objetivo implementar una solución de inteligencia de negocios en la nube que optimice la generación de reportes para el seguimiento de microcréditos en Holding Waki SAC.

El documento se organiza en tres capítulos y una sección final. El **Capítulo 1** presenta los objetivos, el alcance, la situación actual de la empresa y los fundamentos teóricos del trabajo. El **Capítulo 2** desarrolla la arquitectura, el modelo dimensional, el proceso ETL, los dashboards implementados y la validación funcional. El **Capítulo 3** compara las métricas de tiempo, oportunidad, exactitud y retrabajo antes y después de la implementación. Finalmente, las **Conclusiones y Recomendaciones** sintetizan los resultados y las líneas de trabajo futuro. La documentación técnica del sistema se incluye en el **Anexo A**, junto con las actas de conformidad y las pruebas unitarias del ETL en los anexos siguientes.



CAPÍTULO I

DESCRIPCIÓN DEL PROYECTO

1.1. Objetivos

1.1.1. General

Implementar una solución de inteligencia de negocios que optimice la generación de reportes para el seguimiento de microcréditos en una empresa financiera ubicada en Arequipa.

1.1.2. Específicos

1. Diseñar e implementar un proceso ETL en la nube que permita la extracción, transformación y carga de datos desde las fuentes operacionales de la empresa.
2. Desarrollar un Data Warehouse en la nube que consolide los datos operativos y gerenciales de los microcréditos.
3. Crear dashboards interactivos a partir del Data Warehouse en la nube que permitan visualizar métricas clave.
4. Evaluar el impacto de la solución implementada mediante la comparación de métricas de tiempo, oportunidad, exactitud y retrabajo sobre el proceso, antes y después de la implementación.

1.2. Alcances y limitaciones

El presente proyecto se enfocará en el diseño e implementación de una solución de inteligencia de negocios para mejorar el seguimiento de microcréditos en la empresa Holding Waki S.A.C. El alcance incluye: la extracción de datos desde la base Cassandra, el desarrollo de un proceso ETL en AWS Lambda, la carga de información en un Data Warehouse en Amazon Redshift y la creación de dashboards en Power BI para tres usuarios principales: administradora, analista de créditos y junta de socios.

La solución permitirá consultar la información desde cualquier dispositivo con conexión a internet y se limitará únicamente a los datos operativos propios de la empresa, el proyecto no contempla modificaciones al sistema transaccional ni ampliaciones funcionales de Cassandra.

Entre las limitaciones se consideran la dependencia de servicios en la nube (AWS y Power BI), la necesidad de una conexión estable a internet y los costos asociados al consumo de estos servicios. En términos de viabilidad, se cuenta con acceso a los datos necesarios para completar el desarrollo. La autorización para el uso del nombre de la empresa y de los datos necesarios para el proyecto se presenta en el Anexo E.

1.3. Situación actual y línea base

1.3.1. Descripción de la situación actual

La descripción de la situación actual y la definición de la línea base de las métricas que se presentan en las siguientes subsecciones se construyeron a partir de información proporcionada directamente por el personal de Holding Waki SAC, obtenida mediante reuniones de levantamiento con la administradora (Administración de Holding Waki S.A.C., 2025).

Holding Waki SAC es una empresa financiera ubicada en Arequipa enfocada al otorgamiento de microcréditos a trabajadores de empresas (Holding Waki S.A.C., 2024). En los últimos años viene experimentando un crecimiento notorio en el número de clientes y, por ende, en los créditos otorgados. Para gestionar adecuadamente este aumento, y con el objetivo de mantener un control eficiente de los créditos, la administradora de la empresa debe elaborar dos tipos de reportes: uno denominado reporte operativo y otro conocido como reporte gerencial; ambos reportes se elaboran actualmente utilizando hojas de cálculo en Excel, lo que implica una carga significativa de trabajo manual (Administración de Holding Waki S.A.C., 2025).

El reporte operativo se elabora semanalmente y se comparte con el área de operaciones de la empresa, específicamente con el analista de créditos. Este documento consolida información clave sobre la cartera de créditos de la semana en curso: próximas fechas de vencimiento de cuotas, clientes en mora, detalle de los créditos otorgados y aquellos pendientes de desembolso, entre otros datos relevantes. Su elaboración demanda hasta un día hábil completo de trabajo, debido a la actualización manual de gráficos y fórmulas en Excel (Administración de Holding Waki S.A.C., 2025).

El segundo reporte, de carácter gerencial, se elabora cada diez días con el propósito de informar a la plana gerencial (gerente general, gerente de tecnología, gerente de finanzas) y a los accionistas de la empresa, este incluye datos clave como el número de créditos otorgados por producto, el porcentaje de morosidad y los saldos de cartera; la elaboración manual de este reporte en Excel toma aproximadamente un día y medio (Administración de Holding Waki S.A.C., 2025).

1.3.2. Definición de métricas para línea base

1.3.2.1. Tiempo de elaboración de reportes

Las Tablas 1 y 2 evidencian la carga operativa que soporta la administradora para la elaboración de los reportes; la elaboración del reporte operativo requiere en promedio 1 día hábil, y la preparación del reporte gerencial, 1.5 días hábiles (Administración de Holding Waki S.A.C., 2025). Esta situación limita la capacidad de la administradora para atender otras responsabilidades propias de su rol y afecta la disponibilidad de información para la toma de decisiones.

Tabla 1: Análisis de tiempos para la elaboración del Reporte Operativo

Actividad	Tiempo
Elaboración Reporte Operativo (1 día hábil)	
Extraer manualmente la información de microcréditos desde la plataforma	60 minutos
Registrar y estructurar los datos en las hojas de cálculo	80 minutos
Validar la información con contratos enviados por correo electrónico	120 minutos
Validar diariamente la cartera de créditos	120 minutos
Consolidar y dar formato final al reporte operativo	100 minutos
Total	480 minutos (1 día)

Nota. Elaboración propia.

Tabla 2: Análisis de tiempos para la elaboración del Reporte Gerencial

Actividad	Tiempo
Elaboración Reporte Gerencial (1.5 días hábiles)	
Extraer manualmente la información de microcréditos desde la plataforma	120 minutos
Registrar y estructurar los datos en la hoja de cálculo	130 minutos
Validar la información con contratos enviados por correo electrónico	160 minutos
Validar la cartera de créditos del período	180 minutos
Consolidar y dar formato final al reporte gerencial	130 minutos
Total	720 minutos (1.5 días)

Nota. Elaboración propia.

A diferencia del reporte operativo, la elaboración del reporte gerencial cubre un período de 10 días, durante los cuales los estados de los créditos se actualizan de forma continua en la plataforma. Esto obliga a la administradora a realizar la extracción de datos desde cero en cada ciclo, con el fin de incluir créditos que pudieron cambiar de estado, cuotas nuevas generadas y

pagos registrados en ese intervalo, lo que explica el mayor tiempo requerido para las actividades de extracción y validación (Administración de Holding Waki S.A.C., 2025).

1.3.2.2. Oportunidad de la información

La oportunidad de la información se refiere a la disponibilidad de los reportes en el momento en que son requeridos para sostener las decisiones del área. Cuando un reporte llega después del momento esperado, las decisiones que dependen de él se posponen o se toman con información desactualizada; es decir, si el reporte operativo se espera un lunes y recién se entrega el miércoles, durante esos dos días el analista de créditos pierde, como indica la métrica, la oportunidad de tomar decisiones basadas en datos actualizados. La situación es similar con el reporte gerencial: en ambos casos se pierde la oportunidad de tomar decisiones sobre cobranza, publicidad, entre otros aspectos del negocio.

La elaboración de los reportes es una tarea manual que recae exclusivamente sobre la administradora debido a ello cualquier actividad propia de su rol como atender consultas, coordinar con el área operativa o resolver situaciones puntuales del negocio interrumpe el proceso y genera retrasos en la entrega (Administración de Holding Waki S.A.C., 2025). Este patrón se evidencia en el análisis de los últimos 3 meses, elaborado a partir del historial de entregas registrado en los correos electrónicos internos de la empresa (Holding Waki S.A.C., 2025).

Para dimensionar la magnitud del problema, el análisis considera tres elementos: el **ciclo esperado** del reporte, entendido como el período regular entre una entrega y la siguiente según la planificación (semanal para el reporte operativo y cada 10 días para el gerencial); el **retraso promedio**, calculado como la diferencia en promedio en días hábiles entre la fecha esperada y la fecha real de entrega durante el período evaluado; y la **ventana de decisión perdida**, que expresa en porcentaje qué proporción del ciclo esperado transcurre sin información disponible, la misma se calcula como la proporción entre el retraso promedio y el ciclo esperado ($\text{retraso} \div \text{ciclo}$). La Tabla 3 consolida estas tres mediciones.

Tabla 3: Análisis de oportunidad en la entrega de reportes (últimos 3 meses)

Reporte	Ciclo esperado	Retraso promedio	Ventana de decisión perdida
Operativo	7 días	2.3 días hábiles	32.9 % del ciclo
Gerencial	10 días	3.5 días hábiles	35.0 % del ciclo

Nota. Elaboración propia.

Los resultados indican que, en el escenario previo, aproximadamente un tercio de cada ciclo transcurría sin información actualizada disponible para los usuarios, lo que limita la ca-

pacidad del área para reaccionar a tiempo sobre la cartera de créditos y distintos aspectos del negocio.

1.3.2.3. Exactitud y consistencia de la información

Se analizó la calidad de la información en los reportes elaborados durante los últimos 3 meses, a partir de un muestreo de 12 reportes operativos y 9 reportes gerenciales, se identificaron tres categorías de errores: transcripción, cálculo y omisión (Holding Waki S.A.C., 2025).

Como se detalla en la Tabla 4, los resultados revelan que el 25.0 % de los reportes operativos y el 33.3 % de los reportes gerenciales contienen al menos un error, especialmente preocupante es la alta tasa de errores de cálculo (22.2 %) en los reportes gerenciales, clasificados como de alto impacto por afectar directamente decisiones financieras y estratégicas.

Tabla 4: Análisis de exactitud de la información en reportes (últimos 3 meses)

Tipo de error	Rep. operativos	Rep. gerenciales	Impacto
Transcripción	16.7 %	11.1 %	Medio-Alto
Cálculo	8.3 %	22.2 %	Alto
Omisión	8.3 %	11.1 %	Medio
Total con error	25.0 %	33.3 %	Alto

Nota. Elaboración propia.

1.3.2.4. Retrabajo sobre el proceso

Como consecuencia directa de los niveles de error identificados en la subsección anterior, la administradora solicitaba al asistente del área una revisión adicional de cada reporte antes de distribuirlo, con el objetivo de detectar y corregir inconsistencias en los datos transcritos y en los cálculos manuales (Administración de Holding Waki S.A.C., 2025). Esta revisión se ejecutaba después de cada ciclo de elaboración, tanto para el reporte operativo (semanal) como para el gerencial (cada diez días), lo que introduce un componente de retrabajo que no se contempla en las métricas anteriores y que se traduce en horas adicionales de trabajo y, por consiguiente, en un costo económico.

Para cuantificar este retrabajo, se estimó el tiempo dedicado por el asistente a revisar cada reporte como el 75 % del tiempo de elaboración original, considerando que la tarea de revisión implica recorrer la totalidad del documento para verificar cifras y fórmulas, aunque sin reconstruirlo desde cero. La Tabla 5 consolida el cálculo del retrabajo mensual.

Tabla 5: Análisis de retrabajo sobre el proceso de elaboración de reportes

Reporte	Frecuencia	Tiempo de revisión	Reportes/mes	Horas/mes
Operativo	Semanal	6 h (75 % de 8 h)	4	24
Gerencial	Cada 10 días	9 h (75 % de 12 h)	3	27
Total				51 h/mes

Nota. Elaboración propia.

Para traducir las horas de retrabajo a un costo económico, se consideró un sueldo mensual de S/. 1 500 para el asistente del área (Administración de Holding Waki S.A.C., 2025). Asumiendo una jornada laboral estándar de 40 horas semanales (160 horas al mes), la tarifa efectiva resulta en S/. 9.38 por hora. Aplicando esta tarifa sobre las 51 horas mensuales de retrabajo, se obtiene un costo mensual de S/. 478.13, equivalente a aproximadamente S/. 5 737.50 anuales dedicados exclusivamente a revisar los reportes elaborados manualmente.

Tabla 6: Costo económico del retrabajo (línea base)

Concepto	Valor
Sueldo mensual del asistente	S/. 1 500.00
Jornada laboral mensual	160 horas
Tarifa efectiva por hora	S/. 9.38
Horas mensuales de retrabajo	51 horas
Costo mensual del retrabajo	S/. 478.13
Costo anual del retrabajo	S/. 5 737.50

Nota. Elaboración propia.

Esta métrica complementa a las anteriores al mostrar que los errores en los reportes no solo generan riesgo sobre la toma de decisiones, sino que además consumen horas remuneradas del área que podrían destinarse a actividades de mayor valor.

1.4. Fundamentos Teóricos

1.4.1. Antecedentes del proyecto

Damiano Espinoza (2020) construyó un data mart en el Banco Ripley para mejorar la toma de decisiones en el área de Continuidad del Negocio, siguiendo el enfoque bottom-up de la metodología Kimball, el proyecto incluyó el desarrollo de procesos ETL para la integración de datos y la generación de KPIs mediante Power BI, el contexto financiero y el uso de KPIs guardan estrecha relación con los objetivos del presente proyecto, donde también se busca consolidar información crediticia para apoyar la toma de decisiones gerencial y operativa en Holding Waki SAC.

Retuerto y cols. (2023) implementaron inteligencia de negocios con Power BI en sistemas bancarios peruanos, evaluando la metodología Kimball frente a Inmon mediante la técnica de juicio de expertos, resultado en el que Kimball obtuvo mayor puntaje; este antecedente aporta evidencia de la aplicabilidad de Kimball en el sector financiero peruano, contexto compartido con el presente proyecto, y refuerza la elección metodológica adoptada para el diseño del Data Warehouse en Holding Waki SAC.

Pamo Ferro (2022) desarrolló una solución BI con metodología Kimball en una empresa comercial peruana, obteniendo una reducción del 99.62% en el tiempo de generación de reportes y un incremento del 61.10% en la cantidad de reportes producidos. Estos resultados constituyen una referencia directa para el presente proyecto, dado que el tiempo de elaboración de reportes es uno de los tres indicadores con los que se evaluará el impacto de la solución implementada en Holding Waki SAC.

Peralta Villasante (2021) desarrollaron un data mart para el área de ventas de Transportes Reyna con el objetivo de reducir el tiempo de generación de reportes y mejorar el nivel de servicio, aunque emplearon la metodología HEFESTO en lugar de Kimball, los indicadores de evaluación coinciden con los del presente proyecto, lo que permite establecer una comparación entre enfoques metodológicos distintos orientados a los mismos objetivos de mejora operativa.

En el ámbito local, Ramos Ramos (2023) desarrolló una solución de inteligencia de negocios en una empresa del sur del Perú utilizando la metodología Kimball, un modelo dimensional en esquema estrella, procesos ETL y dashboards en Power BI, alcanzando una mejora del 86.79% en tiempo y costo, este antecedente es particularmente relevante para la presente investigación no solo por la proximidad geográfica con Holding Waki SAC, sino porque el tiempo de elaboración de reportes fue un indicador central en su evaluación, al igual que en el presente proyecto.

Moyano y Molina (2020) implementaron un Data Mart con Power BI para analizar las ventas en los Econegocios Gransol, siguiendo un proceso que abarcó desde el análisis de requerimientos hasta el diseño e implementación del dashboard, si bien el contexto es comercial, su

enfoque en la visualización de información para la toma de decisiones operativas es comparable al propósito de los dashboards desarrollados en el presente proyecto para el analista de créditos y la administradora de Holding Waki SAC.

Daza, De la Paz Ttito, y Cajas (2024) implementó una solución BI con Kimball en una empresa de seguridad, estructurando un modelo con 9 dimensiones y una tabla de hechos, y logrando mejorar el nivel de servicio de 49.19% a 77.61% y reducir en 25% el índice de incidencias; este trabajo ofrece una referencia concreta sobre el diseño del modelo dimensional, aspecto directamente aplicable al Data Warehouse desarrollado para Holding Waki SAC.

Anugerah (2022) diseñaron e implementaron un data warehouse para el análisis de transacciones de ventas utilizando la metodología Kimball y esquema estrella, permitiendo analizar información por tiempo, producto, sucursal y vendedor mediante Tableau, aunque el sector es distinto, su aporte al diseño dimensional bajo Kimball es aplicable al presente proyecto, donde el esquema estrella también estructura las dimensiones de análisis de los microcréditos en Holding Waki SAC.

Hinojosa Zarate (2023) implementó una solución de inteligencia de negocios en una institución gubernamental empleando Kimball para el diseño del Data Warehouse y Power BI para la visualización. Mediante un diseño pre-experimental, el estudio registró reducciones en tiempo y costos operativos, este antecedente es relevante porque comparte el enfoque de medición pre-post utilizado en el presente proyecto para evaluar el impacto de la solución en los tres indicadores definidos para Holding Waki SAC.

Vinoth y cols. (2022) analizaron la adopción de cloud computing en el sector bancario, destacando beneficios como escalabilidad y reducción de costos operativos, pero también riesgos asociados a la dependencia de proveedores y la pérdida de control sobre los datos; sus conclusiones son directamente aplicables al presente proyecto, que hace uso de servicios en la nube como AWS Lambda y Amazon Redshift, riesgos que fueron reconocidos explícitamente dentro de las limitaciones del alcance.

Kanchepu (2023) destacan que el cloud computing permite a las instituciones bancarias mejorar su eficiencia operativa, reducir costos y automatizar tareas rutinarias, este enfoque es coherente con la motivación del presente proyecto, donde la adopción de servicios en la nube busca precisamente eliminar el procesamiento manual de datos que actualmente consume hasta día y medio de trabajo en la elaboración de reportes en Holding Waki SAC.

Berisha, Mëziu, y Shabani (2022) comparan los paradigmas ETL y ELT en entornos cloud, señalando que trasladar la transformación de datos directamente a la nube reduce la carga en sistemas locales y acelera el análisis, si bien el presente proyecto utiliza AWS en lugar de las herramientas analizadas por los autores, este marco conceptual fundamenta la decisión de implementar el proceso ETL sobre AWS Lambda con carga hacia Amazon Redshift como Data Warehouse central.

Luengo Herrero y cols. (2023) desarrolló una aplicación ETL/ELT en la nube para monitorizar la productividad en una Smart Factory, utilizando Azure Data Factory, Azure SQL Server y Power BI para la obtención de indicadores en tiempo real. Aunque el dominio es industrial, la arquitectura cloud empleada es comparable a la del presente proyecto, donde AWS Lambda, Redshift y Power BI cumplen roles equivalentes en la cadena de procesamiento de datos de Holding Waki SAC.

Chávez Rengifo y Amaya Pacheco (2023) implementaron una solución de Big Data para el área de comercialización de una empresa peruana utilizando Azure HDInsight, Azure Data Factory, Azure Synapse Analytics y Power BI, logrando centralizar grandes volúmenes de datos para mejorar la toma de decisiones, este caso refuerza la viabilidad de adoptar arquitecturas basadas en servicios cloud administrados para la integración y análisis de datos, enfoque que el presente proyecto replica sobre la infraestructura de AWS.

Apolaya Segura y Flores Suárez (2022) digitalizaron una sala situacional para el monitoreo de COVID-19 en EsSalud Lambayeque empleando Power BI, Azure SQL Database y la metodología Kimball, con acceso a información desde cualquier dispositivo conectado a internet, este antecedente es tecnológicamente uno de los más cercanos al presente proyecto, pues comparte elementos clave de la solución implementada en Holding Waki SAC: una base de datos en la nube, dashboards en Power BI y el acceso remoto como requisito funcional para los usuarios.

1.4.2. Bases Teóricas del proyecto

1.4.2.1. Inteligencia de Negocios

La inteligencia de negocios (Business Intelligence) se refiere al uso de tecnologías, aplicaciones y prácticas para la recolección, integración, análisis y presentación de información empresarial; el objetivo de la BI es apoyar la toma de decisiones empresariales más informada y eficiente mediante la provisión de datos procesables (Frankenfield, 2024).

1.4.2.2. Metodología de Ralph Kimball

La metodología de Ralph Kimball es un enfoque utilizado para el diseño de data warehouses e iniciativas de inteligencia de negocios, se basa en el modelado dimensional, el desarrollo de data marts orientados a procesos del negocio y un enfoque bottom-up, donde las áreas funcionales se integran mediante dimensiones conformadas (Group, 2023).

1.4.2.3. Data Warehouse

Un data warehouse es un sistema centralizado de almacenamiento de datos utilizado para la consolidación de datos provenientes de diversas fuentes, este sistema almacena datos históricos y actuales y está diseñado para facilitar el análisis y la toma de decisiones (Pratt, 2023).

1.4.2.4. Tabla de Hechos

Una tabla de hechos es aquella que contiene los datos cuantitativos (mediciones) sobre el negocio, estas tablas se componen de claves que se unen a las Tablas Dimensionales, y las cantidades numéricas que se quieren analizar (Group, 2008).

1.4.2.5. Tablas Dimensionales

Las tablas dimensionales contienen atributos que describen los objetos de negocio en un Data Warehouse; proporcionan el contexto necesario para interpretar los datos en las Tablas de Hechos, por ejemplo, describiendo productos, clientes o tiempo (Group, 2024).

1.4.2.6. ETL

ETL es un proceso en la integración de datos que implica la extracción de datos de diversas fuentes, la transformación de los datos en un formato adecuado y su carga en un sistema de almacenamiento, como un Data Warehouse (IBM, 2024).

1.4.2.7. Amazon Redshift

Amazon Redshift es un servicio de almacenamiento analítico en la nube que funciona como un Data Warehouse, utilizando arquitectura columnar y procesamiento paralelo para ejecutar consultas de alto rendimiento sobre grandes volúmenes de datos (Amazon Web Services, 2025b).

1.4.2.8. Amazon EventBridge

Amazon EventBridge es un servicio de orquestación basado en eventos que permite programar, automatizar y coordinar la ejecución de procesos en la nube, activando funciones o flujos de trabajo, como tareas ETL en AWS Lambda, según reglas definidas o en intervalos programados (Amazon Web Services, 2025a).

1.4.2.9. AWS Lambda

AWS Lambda es un servicio de computación sin servidor que permite ejecutar funciones en la nube bajo demanda, sin necesidad de aprovisionar o administrar infraestructura; este servicio resulta ideal para la construcción de procesos de extracción, transformación y carga de datos, activándose únicamente cuando se requiere procesar información proveniente de diferentes fuentes y enviándola posteriormente a un sistema de almacenamiento o un Data Warehouse (Amazon Web Services, 2025c).

1.4.2.10. Microsoft PowerBI

Power BI es una plataforma de análisis de datos de Microsoft que proporciona herramientas para la agregación, análisis, visualización y uso compartido de datos, es conocida por sus capacidades de crear cuadros interactivos e informes a partir de diversas fuentes de datos (Microsoft, 2025).

1.4.3. Técnicas y Herramientas

Para el desarrollo del presente proyecto se evaluaron distintas metodologías de diseño de data warehouses, servicios en la nube para la construcción del proceso ETL y herramientas de visualización, el objetivo de esta sección es presentar de manera resumida las alternativas consideradas, junto con una justificación técnica de las tecnologías seleccionadas.

1.4.3.1. Comparación de metodologías para el diseño del Data Warehouse

Se analizaron dos metodologías ampliamente utilizadas para el diseño de data warehouses: el enfoque Bottom-Up de Ralph Kimball y el enfoque Top-Down de Bill Inmon.

Tabla 7: Comparación de metodologías de diseño de Data Warehouse

Metodología	Características	Ventajas	Limitaciones
Ralph Kimball	Enfoque Bottom-Up, basado en modelado dimensional y datamarts.	Fácil implementación, orientada al negocio, ideal para proyectos acotados.	No prioriza una integración corporativa total.
Bill Inmon	Enfoque Top-Down, desarrollo de un EDW corporativo previo a datamarts.	Arquitectura estandarizada y visión global del negocio.	Mayor complejidad, costos y tiempo de implementación.

Nota. Elaboración propia.

La metodología de Ralph Kimball fue seleccionada porque se adapta mejor a proyectos con alcance delimitado, facilita la construcción de un único datamart y permite obtener resultados funcionales en menores tiempos. Su orientación al negocio y simplicidad metodológica la convierten en la opción adecuada para este proyecto.

1.4.3.2. Comparación de herramientas para el desarrollo del ETL

Se evaluaron distintas alternativas para implementar el proceso de extracción, transformación y carga en un entorno cloud, considerando costos, facilidad de integración y nivel de complejidad requerido por el proyecto.

Tabla 8: Comparativa de herramientas ETL en la nube

Herramienta	Tipo	Ventajas	Limitaciones
AWS Lambda	Funciones serverless bajo demanda.	Costo muy bajo (pago por invocación), escalabilidad automática y simplicidad operativa.	Límite de tiempo de ejecución y memoria; orientado a cargas pequeñas o medianas.
AWS Glue	Servicio ETL administrado.	Ideal para grandes volúmenes, dispone de crawlers y administración automática de metadatos.	Costoso para cargas pequeñas; introduce sobreingeniería para el alcance del proyecto.
Azure Data Factory	Servicio ETL visual administrado.	Buena integración dentro del ecosistema Microsoft.	No pertenece al ecosistema AWS planteado y requiere componentes adicionales para integrarse.

Nota. Elaboración propia.

Tras el análisis, se seleccionó **AWS Lambda** debido a que los volúmenes de datos son reducidos y el modelo de cobro por invocación hace que su costo sea significativamente menor frente a alternativas como AWS Glue, cuyo precio se basa en DPU por hora y resulta poco eficiente para cargas ligeras. Además, Lambda se integra de manera nativa con Amazon S3, Amazon Redshift y EventBridge, lo que permite construir un proceso ETL completamente serverless y alineado con la arquitectura planteada, por su parte, Azure Data Factory se descartó al no formar parte del ecosistema AWS ni aportar ventajas reales para el alcance del proyecto.

1.4.3.3. Comparación de tecnologías para el almacenamiento analítico

Como sistema de almacenamiento para el Data Warehouse se evaluaron distintas opciones dentro del ecosistema AWS.

Tabla 9: Comparativa de tecnologías de almacenamiento analítico

Tecnología	Tipo	Ventajas	Limitaciones
Amazon Redshift Serverless	Data warehouse columnar serverless.	Pago por uso, ideal para análisis y BI, escalabilidad automática.	No orientado a cargas transaccionales.
Amazon RDS	Base de datos relacional.	Adecuado para OLTP y sistemas operacionales.	No optimizado para consultas analíticas.
Amazon Athena + S3	Consulta serverless sobre archivos.	Muy bajo costo, sin servidores.	Menor rendimiento que Redshift para modelos dimensionales.

Nota. Elaboración propia.

Se seleccionó **Amazon Redshift Serverless** por ser un Data Warehouse real, optimizado para modelos dimensionales y consultas analíticas, ofreciendo una arquitectura serverless que reduce costos y mantenimiento.

1.4.3.4. Herramientas de visualización

Finalmente, se evaluaron alternativas de herramientas BI para la visualización de los datos consolidados en el Data Warehouse.

Tabla 10: Comparación de herramientas BI

Herramienta	Ventajas	Limitaciones
Power BI	Bajo costo, interfaz intuitiva, gran integración con múltiples fuentes.	Requiere licencia Pro para compartir dashboards.
Tableau	Visualizaciones avanzadas, alta flexibilidad.	Costos significativamente más altos.
Looker Studio	Gratuito y basado en web.	Menos robusto para modelos dimensionales complejos.

Nota. Elaboración propia.

Se eligió **Power BI** debido a su facilidad de uso, accesibilidad económica y capacidad para generar dashboards operativos y gerenciales adecuados para los usuarios de la empresa.



CAPÍTULO II

DESARROLLO DE LA SOLUCIÓN DE INTELIGENCIA DE NEGOCIOS

2.1. Visión general del ciclo de vida y metodología

2.1.1. Justificación del ciclo de vida seleccionado

El ciclo de vida adoptado para el desarrollo del Data Warehouse se basa en la metodología propuesta por Kimball, Ross, Thornthwaite, Mundy, y Becker (2008), la cual plantea un enfoque bottom–up orientado al negocio, sustentado en modelos dimensionales y en la construcción incremental de *datamarts*, esto quiere decir que el desarrollo no parte de un diseño abstracto, sino de las necesidades del negocio, cada área o proceso se aborda de manera independiente, construyendo pequeños *datamarts* que luego se integran entre sí. Este enfoque permite entregar valor más rápido, adaptar el modelo a cambios en los procesos y asegurar que las métricas respondan directamente a cómo la organización toma decisiones en la práctica, la metodología estructura el ciclo de vida en fases definidas: levantamiento de requerimientos de negocio, diseño dimensional, diseño físico, desarrollo de procesos ETL, implementación y despliegue de reportes.

Frente a otras alternativas, como el enfoque top–down propuesto por Inmon, Strauss, y Neushloss (2010), basado en un Enterprise Data Warehouse normalizado, la metodología de Kimball presenta ventajas relevantes para el contexto de Holding Waki SAC, al tratarse de una *fintech* de tamaño reducido y con recursos limitados, resulta crítico disponer de un ciclo de vida que permita entregas rápidas y continuas, además, la naturaleza del problema (seguimiento operativo y gerencial de microcréditos) exige una estructura de datos optimizada para consultas analíticas, lo cual se adapta de manera ideal al modelado dimensional sobre Amazon Redshift Serverless e integra de forma sencilla con Power BI; por estas razones, la elección del ciclo de vida de Kimball resulta adecuada para el contexto de Holding Waki SAC, al reducir el tiempo de entrega de resultados, reducir la complejidad de implementación inicial y mantener una orientación a las necesidades de negocio en el seguimiento de microcréditos.

2.1.2. Adaptación de la metodología al proyecto

La metodología de Kimball fue adaptada al tamaño de la empresa y al alcance del proyecto mediante el enfoque a un único proceso de negocio: seguimiento de microcréditos. En la etapa de levantamiento de requerimientos se priorizaron únicamente los indicadores operativos y gerenciales solicitados por la administradora, el analista de créditos y la junta de socios (Administración de Holding Waki S.A.C., 2025); dejando de lado procesos no relevantes, esto permitió definir el alcance del primer *datamart* de microcréditos, enfocado en el ciclo de vida de un microcrédito y sus cuotas.

El modelo dimensional se define como un esquema estrella, con una tabla de hechos de microcréditos y un conjunto de dimensiones principales (como cliente y tiempo, etc). Para el proceso ETL, se usará una arquitectura basada en funciones AWS Lambda orquestadas con EventBridge, estas funciones extraerán los datos desde Cassandra, aplicarán las transformaciones requeridas y cargarán tanto las dimensiones como la tabla de hechos en Redshift, por último, la capa de BI se centrará en un conjunto de dashboards en Power BI alineados con los objetivos del proyecto, priorizando la claridad de las métricas y la facilidad de uso para los usuarios.

2.2. Análisis funcional y de datos

2.2.1. Requerimientos de información

Los requerimientos de información se organizaron en función de los tres usuarios principales que utilizan la información: la administradora, el analista de créditos y la junta de socios, durante las entrevistas y el análisis de los reportes actuales, se confirmó que la empresa maneja dos tipos de necesidades claramente diferenciadas (Administración de Holding Waki S.A.C., 2025; Holding Waki S.A.C., 2025).

Por un lado, el equipo operativo requiere información detallada y actualizada sobre el estado diario de los microcréditos (vencimientos próximos, morosidad, desembolsos recientes, etc.); por otro lado, la gerencia y los socios necesitan una vista consolidada que resuma el comportamiento de la cartera en el tiempo, con indicadores globales que apoyen la toma de decisiones estratégicas, dado que esta separación ya existe en los reportes manuales que la empresa elabora actualmente, se mantuvo dicha estructura en el diseño de los dashboards, asegurando que cada grupo de usuarios reciba la información en el nivel de detalle adecuado (Administración de Holding Waki S.A.C., 2025).

En el nivel operativo, tanto la administradora como el analista de créditos necesitan ver el detalle de cada microcrédito y de sus cuotas, lo más consultado diariamente es el listado de créditos vigentes y vencidos, el calendario de pagos por cliente y por fecha, las cuotas que están por vencer, los montos en mora, las reprogramaciones y las cancelaciones, también requieren poder revisar rápidamente la trazabilidad de los pagos realizados (Administración de Holding Waki S.A.C., 2025). A partir de esto se desprenden indicadores operativos como: créditos vigentes, cuotas vencidas, días de atraso por cliente.

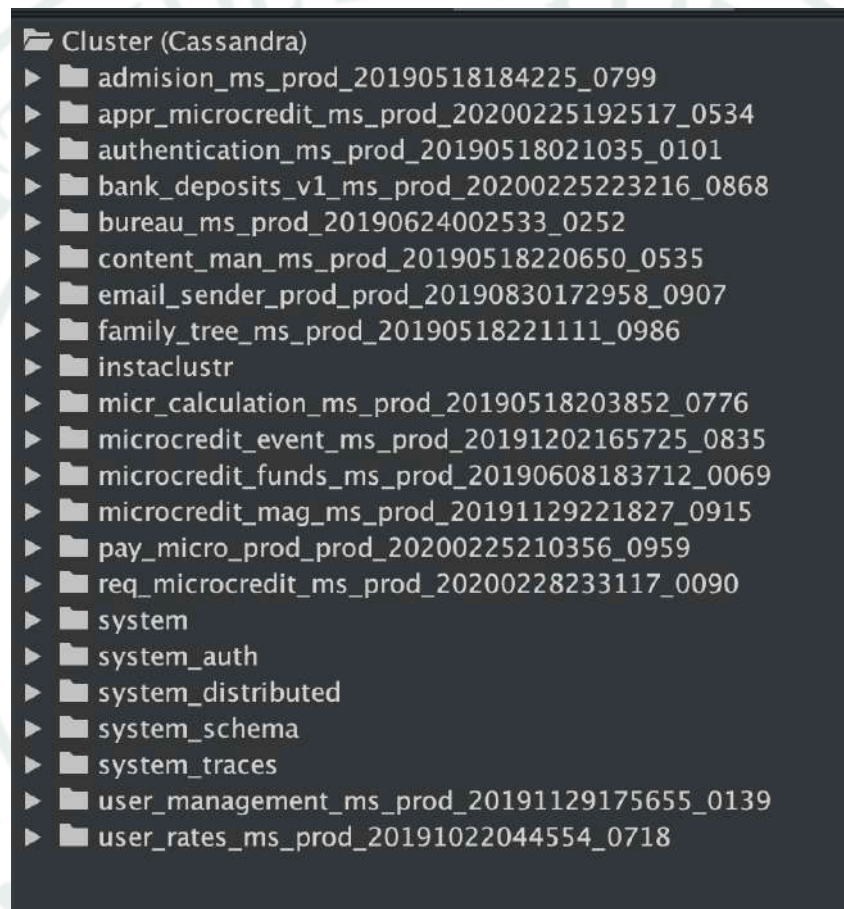
En el nivel gerencial, la junta de socios necesita una vista más resumida que muestre cómo está funcionando la cartera en general y cómo evoluciona el riesgo, para ellos son clave indicadores agregados como el saldo total de la cartera, la tasa de morosidad, las colocaciones por período, la concentración de la cartera según tipo de producto o segmento de cliente, la rotación y el comportamiento histórico de los atrasos, además, requieren analizar estas métricas por distintas dimensiones como tiempo, zona, tipo de cliente, producto para detectar tendencias,

variaciones y oportunidades de mejora (Administración de Holding Waki S.A.C., 2025).

2.2.2. Fuentes de datos y reglas de negocio

Las fuentes de datos de la solución corresponden a los microservicios operacionales desplegados sobre Cassandra, la figura 1 muestra todos los keyspaces presentes en la base de datos de Cassandra:

Figura 1: Keyspaces de los microservicios de Cassandra



Nota. Elaboración propia.

Para la información de clientes se utilizan principalmente los keyspaces `admission_ms` y `user_management_ms`. En el primero, la tabla `payroll` almacena el prerregistro de trabajadores de empresas con convenio para el producto Waki Planilla (datos cargados desde archivos proporcionados por la empresa empleadora), mientras que la tabla `user` de admisión concentra un primer registro de datos personales asociados a un `user_id`. Posteriormente, al completar el registro en la plataforma, la información consolidada del usuario se distribuye en varias tablas del keyspace `user_management_ms`, destacando `user` (datos personales y de contacto), `user_employment` (situación laboral y centro de trabajo), `user_by_dni` y `user_by_email` (índices de búsqueda por DNI y correo). Esta estructura implica que para construir una dimensión

de cliente consistente es necesario integrar registros de admisión y de gestión de usuarios, utilizando como claves de enlace el dni y el user_id.

En el ámbito de los microcréditos y sus pagos, el flujo de datos atraviesa varios keyspaces según la etapa del ciclo de vida del crédito:

1. **Solicitud:** el keyspace req_microcredit_ms contiene la tabla microcredits con los datos de cada solicitud (usuario, DNI, estado civil, score crediticio, período, estado de la solicitud) y tablas auxiliares como amount (monto solicitado, número de cuotas, tasa de interés, TEA, TCEA).
2. **Aprobación:** el keyspace appr_microcredit_ms registra en request_approved las solicitudes aprobadas con la tasa de interés asignada, montos mínimo y máximo, y el historial de créditos previos del usuario.
3. **Desembolso:** en microcredit_mag_ms, las tablas disbursement, disbursement_pendient y disbursement_reviewed registran el proceso de desembolso, incluyendo el monto transferido, banco destino, cuenta del cliente, número de operación y el voucher que evidencia la transferencia.
4. **Crédito activo:** una vez aprobado el desembolso, la información del crédito se consolida en microcredit_mag_ms.microcredits y se replica en vistas auxiliares como microcredit_by_user, microcredit_by_state y microcredit_by_period para facilitar consultas operativas.
5. **Cronograma y cuotas:** la tabla schedule almacena el resumen del cronograma (monto total, número de cuotas, tasa de interés, gastos administrativos, seguro de desgravamen, mora acumulada), mientras que quota_details contiene el detalle de cada cuota (amortización, interés, saldo pendiente) y payment_schedule registra cada cuota programada con su fecha de vencimiento, estado, días de atraso y monto de mora.
6. **Pagos:** la tabla payment registra cada pago realizado por el cliente, incluyendo el monto, banco receptor, número de operación, fecha, cuota asociada (fee_id) y estado de aprobación, las tablas payment_pendient y payment_reviewed gestionan el flujo de revisión de pagos.
7. **Eventos:** el keyspace microcredit_event_ms mantiene en payment_microcredit los eventos asociados a cada cuota (fecha de activación, fecha de vencimiento, estado: Activo, Vencida, Pagada) y en schedule_delayed las cuotas con atraso.

A partir de estos servicios se definen las siguientes reglas de negocio para unificar el ciclo de vida del crédito:

- **Tipo de producto:** se identifica como Waki Planilla si el usuario tiene user_origin y user_origin_app vacíos y existe un registro previo en payroll; en caso contrario se clasifica como Waki Consumo.
- **Estado del crédito:** se determina a partir del campo state en microcredit_mag_ms.microcredits, con valores como Activo, Retrasado o Finalizado.
- **Estado de la cuota:** se obtiene de payment_schedule.state (Pendiente, Finalizado, Vencido) y payment_state (Aprobado, Pendiente, Rechazado).
- **Días de atraso:** se calculan a partir del campo delay en payment_schedule o como la diferencia entre la fecha actual y expiration_date cuando la cuota no ha sido pagada.
- **Monto en mora:** corresponde al campo payment_delay en payment_schedule, que representa el cargo adicional por días de atraso.
- **Consolidación del cliente:** se integran los datos de user_management_ms.user (datos personales), user_employment (datos laborales) y, cuando aplica, admision_ms.payroll (datos de prerregistro), utilizando user_id y dni como claves de enlace.

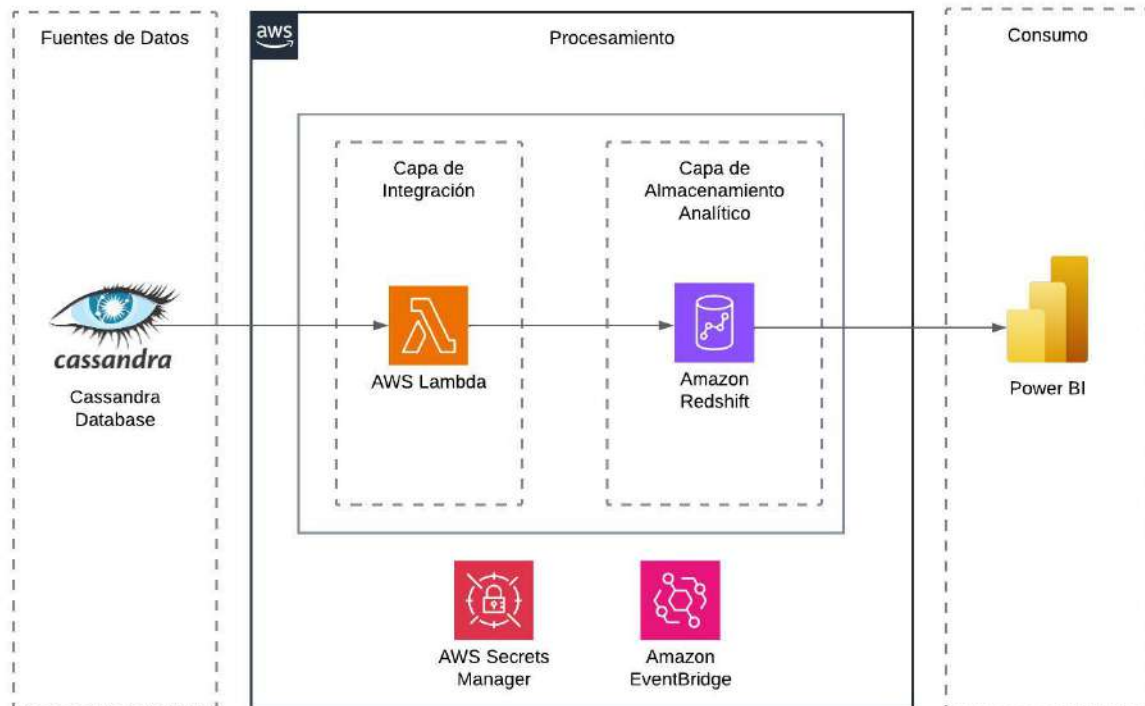
Estas reglas permiten traducir la lógica distribuida de los micros servicios en una vista analítica de créditos, cuotas, pagos y estados de mora en el Data Warehouse.

2.3. Diseño de la arquitectura de la solución

2.3.1. Arquitectura general en la nube

Se presenta en la Figura 2 la arquitectura propuesta en AWS para soportar el proceso analítico, incluyendo la capa de integración, el almacenamiento analítico y la capa de visualización.

Figura 2: Arquitectura del ETL



Nota. Elaboración propia.

2.3.2. Componentes principales

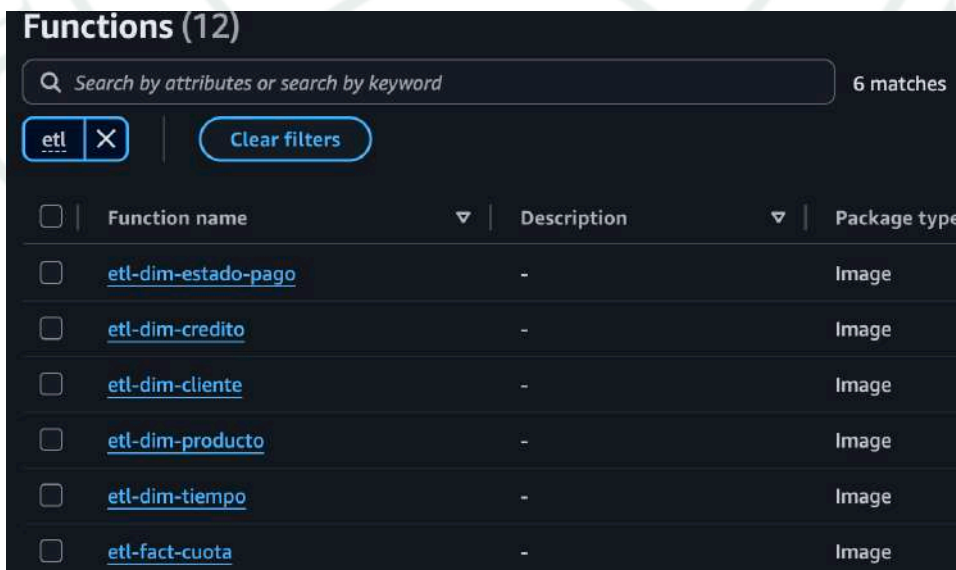
- Apache Cassandra (fuente de datos):** Base de datos NoSQL distribuida que almacena los datos operacionales de los microservicios de Holding Waki SAC, la solución extrae información de múltiples *keyspaces*, incluyendo *user_management_ms* para datos de clientes, *microcredit_mag_ms* para la gestión de créditos activos y cronogramas de pago, y *req_microcredit_ms* para las solicitudes de crédito.
- AWS Lambda:** Servicio de computación *serverless* utilizado para ejecutar las funciones ETL. Se implementaron seis funciones Lambda que comparten una única imagen Docker desplegada en Amazon ECR: tres para dimensiones estáticas (*dim_tiempo*, *dim_estado_pago*, *dim_producto*), dos para dimensiones dinámicas extraídas de Cassandra (*dim_cliente*, *dim_credito*) y una para la tabla de hechos (*fact_cuota*).
- AWS Secrets Manager:** Servicio de gestión de credenciales que almacena de forma segura las contraseñas de Cassandra y Redshift. Las funciones Lambda recuperan las credenciales en tiempo de ejecución mediante el SDK de AWS, evitando almacenar información sensible en variables de entorno o código fuente.
- Amazon Redshift Serverless:** *Data warehouse* columnar en la nube que almacena el modelo dimensional. Se configuró un *workgroup* denominado *waki-bi-workgroup* con

el esquema `dw_microcreditos`, que contiene las cinco tablas dimensionales y la tabla de hechos; el modelo *serverless* permite escalar automáticamente según la demanda de consultas, optimizando costos al facturar únicamente por el procesamiento utilizado.

- **Amazon EventBridge:** Servicio de orquestación basado en eventos que programa la ejecución automática de las funciones ETL. Se configuraron reglas *cron* para ejecutar las cargas dimensionales y de hechos en intervalos definidos, garantizando la actualización periódica del *data warehouse* sin intervención manual.
- **Microsoft Power BI:** Plataforma de inteligencia de negocios utilizada para la capa de visualización. Dado que Amazon Redshift Serverless se encuentra en una VPC privada sin acceso público directo, se implementó una arquitectura híbrida de conexión que diferencia entre entornos de desarrollo y producción. En el entorno de desarrollo, Power BI Desktop se conecta mediante túneles SSM (Session Manager) a través de una instancia EC2 que actúa como bastión, mientras que en producción se utiliza el *On-premises Data Gateway* instalado en la misma instancia EC2 para permitir la actualización automática de reportes publicados en Power BI Service. Esta arquitectura garantiza la seguridad de los datos al no exponer puertos públicos de Redshift, manteniendo el acceso controlado mediante AWS IAM y Security Groups; los dashboards finales se publican en Power BI Service para acceso compartido entre la administradora, el analista de créditos y la junta de socios.

Las figuras 3 y 4 muestran la configuración de las funciones Lambda y el repositorio de imágenes Docker en la consola de AWS. La figura 5 muestra la configuración de credenciales en AWS Secrets Manager, donde se almacenan de forma segura los accesos a Cassandra y Redshift.

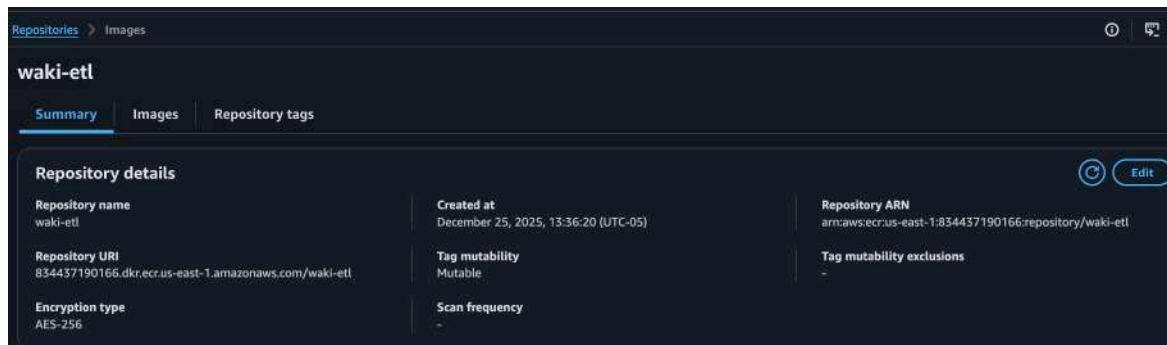
Figura 3: Funciones Lambda del ETL en AWS Console



<input type="checkbox"/>	Function name	Description	Package type
<input type="checkbox"/>	etl-dim-estado-pago	-	Image
<input type="checkbox"/>	etl-dim-credito	-	Image
<input type="checkbox"/>	etl-dim-cliente	-	Image
<input type="checkbox"/>	etl-dim-producto	-	Image
<input type="checkbox"/>	etl-dim-tiempo	-	Image
<input type="checkbox"/>	etl-fact-cuota	-	Image

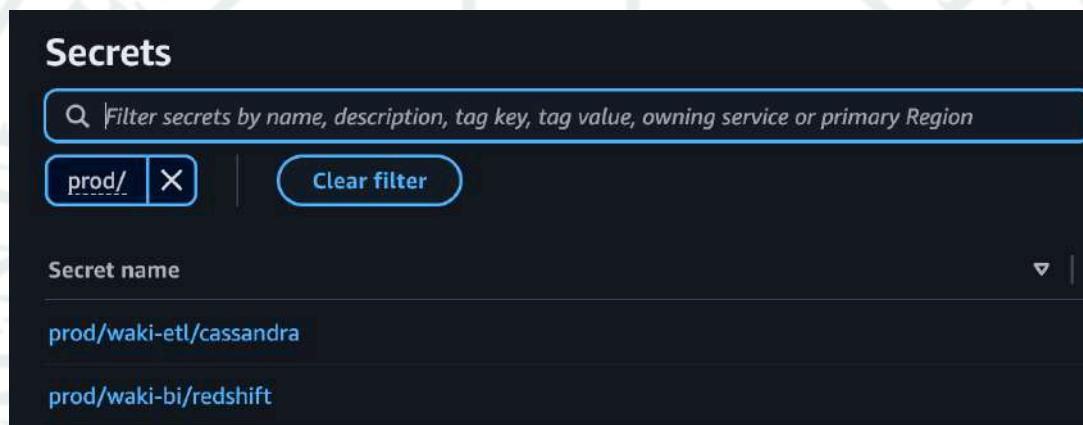
Nota. Elaboración propia.

Figura 4: Repositorio ECR con imagen Docker del ETL



Nota. Elaboración propia.

Figura 5: Credenciales almacenadas en AWS Secrets Manager



Nota. Elaboración propia.

2.3.3. Flujos de datos y orquestación

El flujo de datos de la solución sigue un proceso ETL (*Extract, Transform, Load*) ejecutado de manera automatizada cada día. El proceso se divide en tres fases principales:

1. **Extracción:** Las funciones Lambda se conectan a los *keyspaces* de Cassandra mediante consultas CQL; para las dimensiones dinámicas (*dim_cliente*, *dim_credito*) y la tabla de hechos (*fact_cuota*), se extraen los registros de las tablas operacionales *user*, *user_employment*, *microcredits*, *payment_schedule*, *quota_details* y *schedule*; las dimensiones estáticas (*dim_tiempo*, *dim_estado_pago*, *dim_producto*) se generan directamente en código Python sin acceder a Cassandra.
2. **Transformación:** Los datos extraídos se procesan utilizando la librería Pandas; las transformaciones incluyen: conversión de identificadores UUID a cadenas de texto, unión (*merge*) de tablas relacionadas, limpieza de valores nulos, cálculo de campos derivados (como días de atraso y banderas de mora), y mapeo de claves sustitutas hacia las dimensiones correspondientes (*sk_producto*, *sk_estado_pago*).

3. **Carga:** Los *DataFrames* resultantes se insertan en Amazon Redshift Serverless mediante el conector `redshift_connector`. Para las cargas completas (*full load*), se ejecuta un TRUNCATE seguido de INSERT; para cargas incrementales, se utiliza un patrón de tabla de *staging* con DELETE y posterior INSERT de los registros actualizados.

2.3.3.1. Orquestación con Amazon EventBridge Scheduler

La ejecución automática del proceso ETL se gestiona mediante Amazon EventBridge Scheduler, que invoca cada función Lambda según un cronograma diario. Los *schedules* están configurados en zona horaria America/Lima (UTC-5), garantizando que las cargas se ejecuten en horarios de baja demanda operacional. La tabla 11 presenta el detalle de la programación:

Tabla 11: Programación de ejecución de procesos ETL

Función Lambda	Expresión cron	Hora (Perú)
etl-dim-tiempo	cron(00 00 * * ? *)	00:00
etl-dim-estado-pago	cron(15 00 * * ? *)	00:15
etl-dim-producto	cron(40 00 * * ? *)	00:40
etl-dim-cliente	cron(00 01 * * ? *)	01:00
etl-dim-credito	cron(30 01 * * ? *)	01:30
etl-fact-cuota	cron(00 02 * * ? *)	02:00

La figura 6 muestra la configuración de los schedules en Amazon EventBridge Scheduler.

Figura 6: Schedules del ETL en Amazon EventBridge Scheduler

Schedule name	Schedule group	Status	Target	Target type
schedule-etl-estado-pago	default	Enabled	etl-dim-estado-pago	LAMBDA_Invoke
schedule-etl-fact-cuota	default	Enabled	etl-fact-cuota	LAMBDA_Invoke
schedule-etl-credito	default	Enabled	etl-dim-credito	LAMBDA_Invoke
schedule-etl-tiempo	default	Enabled	etl-dim-tiempo	LAMBDA_Invoke
schedule-etl-cliente	default	Enabled	etl-dim-cliente	LAMBDA_Invoke
schedule-etl-producto	default	Enabled	etl-dim-producto	LAMBDA_Invoke

Nota. Elaboración propia.

El orden de ejecución respeta las dependencias del modelo dimensional: primero se cargan las dimensiones estáticas, luego las dimensiones dinámicas que alimentan las claves foráneas, y finalmente la tabla de hechos que referencia a todas las dimensiones.

2.3.3.2. Puntos de control y monitoreo

Cada función Lambda registra eventos de *logging* en Amazon CloudWatch Logs, permitiendo rastrear el número de registros extraídos, transformados y cargados en cada ejecución, en caso de error, la función retorna un código de estado HTTP 500 con el detalle del fallo, facilitando el diagnóstico. Adicionalmente, las políticas de reintento de EventBridge Scheduler están configuradas para no reintentar automáticamente (`MaximumRetryAttempts:0`), delegando el manejo de fallos a la revisión manual de los registros de CloudWatch.

2.4. Diseño del modelo dimensional

2.4.1. Definición de la granularidad

El diseño del *datamart* de microcréditos se orienta al seguimiento detallado del comportamiento de pagos y morosidad, por lo que se definió la granularidad de la tabla de hechos a nivel de cuota de microcrédito, es decir, cada fila de la tabla de hechos representa una cuota específica de un microcrédito para un cliente determinado, tal como se encuentra registrada en las estructuras operacionales de cronograma y pagos (`schedule`, `quota_details`, `payment_schedule`, `payment`), esta decisión permite calcular indicadores como número de cuotas vencidas, días de atraso, monto en mora y recuperaciones por período, por cliente o por producto, sin perder el detalle necesario para el análisis operativo.

Se evaluó la alternativa de modelar la tabla de hechos a nivel de microcrédito (una fila por crédito), pero este enfoque no permitía responder con precisión a los requerimientos operativos y gerenciales relacionados con el estado de cada cuota, la identificación de cuotas próximas a vencer y la medición de la recuperación de cartera por pagos parciales. Al fijar la granularidad en la cuota, las métricas a nivel de microcrédito o de cartera (por ejemplo, saldo total, tasa de morosidad, colocaciones por período) se obtienen mediante agregaciones sobre la tabla de hechos, manteniendo un diseño flexible y alineado con las necesidades de seguimiento de la cartera de Holding Waki SAC.

2.4.2. Dimensiones del *datamart* de microcréditos

El *datamart* de microcréditos se estructura en torno a una tabla de hechos central y un conjunto de dimensiones que permiten analizar la cartera desde distintas perspectivas de negocio. Las dimensiones definidas son las siguientes:

- **Dimensión Cliente:** agrupa la información del titular del microcrédito, integrando datos procedentes de los keyspaces `user_management_ms` y `admission_ms`; incluye atributos de identificación (código interno, dni), datos personales (nombres, apellidos, fecha de nacimiento, estado civil), datos de contacto y localización (distrito, provincia, departamento, correo electrónico, teléfono), así como datos laborales (situación laboral, empresa, ingreso mensual cuando está disponible). Esta dimensión permite segmentar la cartera por zona geográfica, condición laboral y perfil del cliente.
- **Dimensión Crédito:** describe las características financieras del microcrédito asociado a cada cuota, se alimenta principalmente de las tablas `microcredit_mag_ms.microcredits`, `amount`, `disbursement`, `schedule` y `req_microcredit_ms.microcredits`. Incluye atributos como identificador del microcrédito (`codigo_credito`), monto solicitado, monto aprobado, número de cuotas, tasa de interés, tasa de mora, gastos administrativos, seguro de desgravamen, fecha de solicitud, fecha de desembolso, estado general del crédito (Activo, Retrasado, Finalizado) y `score_crediticio`. Esta dimensión soporta el análisis de la colocación y comportamiento de la cartera por tipo de crédito.
- **Dimensión Tiempo:** representa el tiempo en sus diferentes niveles de agregación (día, mes, trimestre, año) y se utiliza tanto para la fecha de vencimiento de la cuota como para la fecha de pago efectivo, permite construir indicadores de evolución temporal de la cartera (por ejemplo, tasa de morosidad mensual, recuperaciones por período, comportamiento histórico de atrasos) y se integra naturalmente con los requerimientos de análisis por período definidos para los reportes gerenciales.
- **Dimensión Producto:** clasifica cada relación cliente–crédito en función del tipo de producto y del canal de origen. A partir de la integración de `user_management_ms.user` y `admission_ms.payroll`, se deriva un atributo que distingue entre Waki Planilla (clientes prerregistrados en payroll y sin valores en `user_origin` / `user_origin_app`) y Waki Consumo (clientes que ingresan directamente por la plataforma web o canales digitales), dentro de Waki Planilla se distingue además el subtipo de producto (Crédito por Convenio o Adelanto de Sueldo) según el campo `bureau_msg`, adicionalmente, se incluyen atributos como canal de admisión (`user_origin_app`) y segmento de producto, permitiendo analizar la cartera por tipo de microcrédito y origen comercial.
- **Dimensión Estado de Pago:** recoge los distintos estados en los que puede encontrarse una cuota desde la perspectiva operativa y de riesgo, el estado de la cuota (`payment_schedule`

.state) puede ser Activo, Pendiente, Vencida o Finalizado, mientras que el estado del pago (payment_schedule.payment_state) indica si el pago está Pendiente o Aprobado. Esta dimensión facilita la construcción de indicadores de cartera vigente, en mora y recuperada, en línea con los requerimientos de seguimiento operativo de la administradora.

En conjunto, estas dimensiones cubren las principales preguntas de negocio identificadas: distribución de la cartera por tipo de cliente y producto, comportamiento del crédito a lo largo del tiempo, concentración por zona geográfica y empresa, y estado de pago de las cuotas a nivel operativo y gerencial.

2.4.2.1. Atributos de la Dimensión Cliente

Tabla 12: Atributos de la Dimensión Cliente

Atributo	Tipo	Descripción y justificación
sk_cliente	BIGINT	Clave sustituta secuencial
codigo_usuario	VARCHAR(50)	UUID del usuario en BD fuente (Cassandra)
nombres	VARCHAR(200)	Nombres del cliente
apellidos	VARCHAR(200)	Apellidos del cliente
dni	VARCHAR(20)	Documento de identidad
email	VARCHAR(200)	Correo electrónico
telefono	VARCHAR(50)	Número celular
fecha_registro	DATE	Fecha de alta en la plataforma Waki
tipo_cliente	VARCHAR(50)	Segmentación: Planilla o Consumo
estado_civil	VARCHAR(50)	Estado civil para análisis demográfico
genero	VARCHAR(10)	Género del cliente (M/F)
fecha_nacimiento	DATE	Fecha de nacimiento para edad
distrito	VARCHAR(100)	Distrito de residencia
provincia	VARCHAR(100)	Provincia de residencia
departamento	VARCHAR(100)	Departamento de residencia
tipo_vivienda	VARCHAR(50)	Clasificación de vivienda
ocupacion	VARCHAR(100)	Ocupación o profesión declarada
empresa	VARCHAR(200)	Empresa empleadora
salario_mensual	NUMERIC(18,2)	Ingreso mensual declarado en soles

Nota. Elaboración propia.

2.4.2.2. Atributos de la Dimensión Crédito

Tabla 13: Atributos de la Dimensión Crédito

Atributo	Tipo	Descripción y justificación
sk_credito	BIGINT	Clave sustituta secuencial
codigo_credito	VARCHAR(50)	UUID del microcrédito fuente
monto_solicitado	NUMERIC(18,2)	Monto solicitado por el cliente (soles)
monto_aprobado	NUMERIC(18,2)	Monto finalmente aprobado por Waki
tasa_interes	NUMERIC(8,4)	Tasa de interés mensual aplicada
tasa_mora	NUMERIC(8,4)	Tasa diaria por mora o atraso
plazo_meses	INTEGER	Plazo del crédito en meses
cuota_mensual	NUMERIC(18,2)	Monto estimado de cuota mensual
tea	NUMERIC(8,4)	Tasa Efectiva Anual
tcea	NUMERIC(8,4)	Tasa de Costo Efectivo Anual
gastos_administrativos	NUMERIC(18,2)	Gastos administrativos cobrados
seguro_desgravamen	NUMERIC(18,2)	Prima de seguro de desgravamen
fecha_solicitud	DATE	Fecha en que se solicitó el crédito
fecha_aprobacion	DATE	Fecha de aprobación del crédito
fecha_desembolso	DATE	Fecha de desembolso efectivo
estado_credito	VARCHAR(50)	Estado general del crédito
banco_desembolso	VARCHAR(200)	Banco destino del desembolso
score	NUMERIC(5,2)	Score crediticio del cliente
motivo_rechazo	VARCHAR(500)	Motivo de rechazo si fue rechazado

Nota. Elaboración propia.

2.4.2.3. Atributos de la Dimensión Estado de Pago

Tabla 14: Atributos de la Dimensión Estado de Pago

Atributo	Tipo	Descripción y justificación
sk_estado_pago	INTEGER	Clave sustituta (1-8)
estado_cuota	VARCHAR(50)	Estado de la cuota
estado_pago	VARCHAR(50)	Estado del pago
descripcion	VARCHAR(200)	Estado combinado para reportes

Nota. Elaboración propia.

2.4.2.4. Atributos de la Dimensión Producto

Tabla 15: Atributos de la Dimensión Producto

Atributo	Tipo	Descripción y justificación
sk_producto	INTEGER	Clave sustituta (1-7)
tipo_producto	VARCHAR(50)	Línea de producto
subtipo_producto	VARCHAR(100)	Subtipo específico
user_origen	VARCHAR(100)	Canal de origen del usuario
user_origen_app	VARCHAR(100)	Origen específico en app
descripcion	VARCHAR(200)	Descripción para reportes

Nota. Elaboración propia.

2.4.2.5. Atributos de la Dimensión Tiempo

Tabla 16: Atributos de la Dimensión Tiempo

Atributo	Tipo	Descripción y justificación
sk_tiempo	INTEGER	Clave sustituta en formato YYYYMMDD
fecha	DATE	Fecha completa para joins y filtros
dia	INTEGER	Día del mes (1-31)
mes	INTEGER	Número del mes (1-12)
nombre_mes	VARCHAR(20)	Nombre del mes (Enero, Febrero...)
trimestre	INTEGER	Trimestre del año (1-4)
anio	INTEGER	Año (2019-2026)
dia_semana	INTEGER	Día de la semana
nombre_dia	VARCHAR(20)	Nombre del día
es_fin_semana	BOOLEAN	Indicador de fin de semana
semana_anio	INTEGER	Número de semana del año (1-52)

Nota. Elaboración propia.

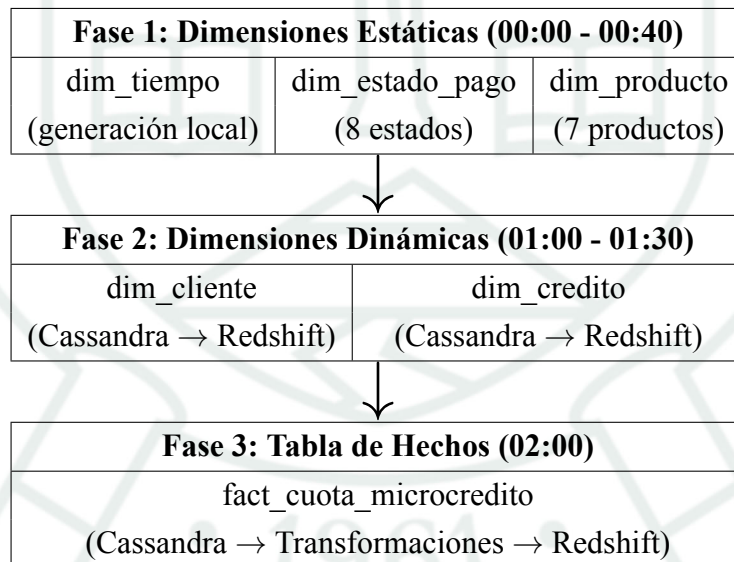
2.4.3. Flujo de datos del proceso ETL

El flujo se divide en tres fases:

1. **Fase 1 - Dimensiones Estáticas:** Se generan las dimensiones que no dependen de datos fuente externos, la dimensión tiempo se carga con fechas desde 2019 hasta 2026, mientras que las dimensiones producto y estado de pago contienen valores predefinidos según las reglas de negocio.
2. **Fase 2 - Dimensiones Dinámicas:** Se extraen y transforman los datos de clientes y créditos desde Cassandra, estas dimensiones dependen de múltiples tablas fuente y requieren operaciones de *merge* para consolidar la información.
3. **Fase 3 - Tabla de Hechos:** Una vez cargadas todas las dimensiones, se procesa la tabla de hechos que referencia a las cinco dimensiones mediante claves sustitutas, esta fase incluye el cálculo de campos derivados y la asignación de las claves foráneas correspondientes.

El proceso ETL sigue un flujo secuencial que garantiza la integridad referencial del modelo dimensional. La figura 7 ilustra el orden de ejecución y las dependencias entre las etapas:

Figura 7: Flujo de datos del proceso ETL



Nota. Elaboración propia.

2.4.4. Tabla de hechos

La tabla de hechos del *datamart* de microcréditos, denominada *fact_cuota_microcredito*, almacena una fila por cada cuota programada de un microcrédito. Cada registro se construye a partir de la integración de la información de cronograma (*schedule*, *quota_details*, *payment_schedule*), pagos (*payment*) y eventos de cuota (*payment_microcredit*, *schedule_delayed*), enlazando con las dimensiones mediante claves sustitutas. La tabla de hechos incluye las siguientes claves foráneas:

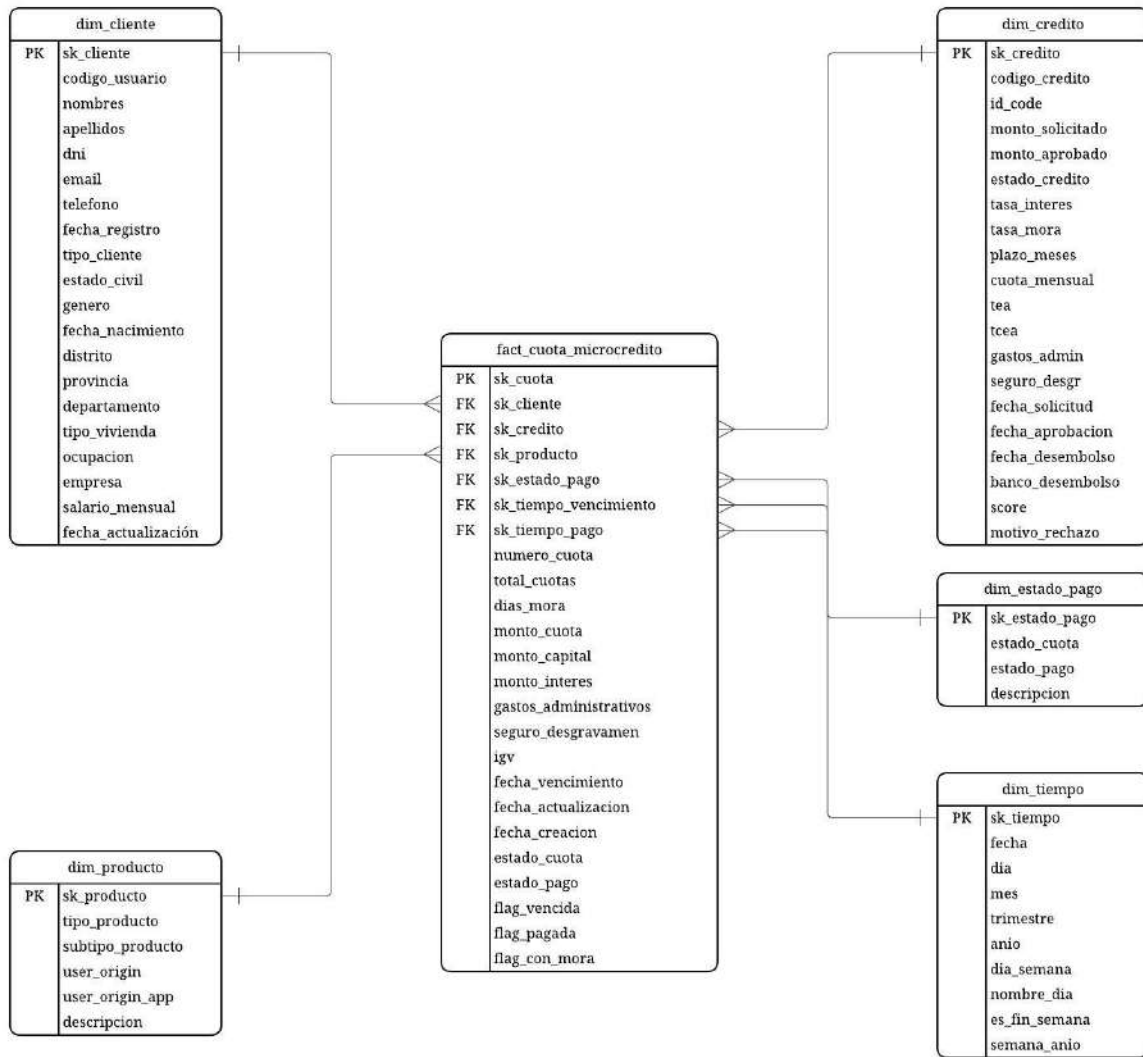
- Clave foránea hacia la **Dimensión Cliente** (cliente titular de la cuota).
- Clave foránea hacia la **Dimensión Crédito** (microcrédito al que pertenece la cuota).
- Claves foráneas hacia la **Dimensión Tiempo** para la fecha de vencimiento y, cuando existe, la fecha de pago efectivo.
- Clave foránea hacia la **Dimensión Producto** (Waki Planilla o Waki Consumo, canal de origen).
- Clave foránea hacia la **Dimensión Estado de Pago** (situación actual de la cuota).

Además de estas claves, la tabla de hechos contiene las medidas cuantitativas necesarias para el análisis de la cartera, derivadas de los montos y estados presentes en las tablas operacionales. Entre las medidas más relevantes se encuentran: monto programado de la cuota (incluyendo capital, intereses y gastos), monto efectivamente pagado, saldo pendiente de la cuota, monto de mora asociado, días de atraso, número de cuota dentro del crédito, número total de cuotas del microcrédito y banderas lógicas para identificar si la cuota está vencida, pagada o en proceso de regularización. Estas medidas son agregables a lo largo de las dimensiones cliente, tiempo y producto, y permiten obtener tanto indicadores operativos de detalle (por ejemplo, cuotas vencidas por cliente en una fecha determinada) como indicadores agregados de gestión (saldo total de la cartera, tasa de morosidad, recuperación por período y por tipo de producto) a partir de la misma estructura de hechos.

2.4.5. Diagrama del modelo dimensional

La figura 8 presenta el esquema estrella del *datamart* de microcréditos, mostrando la tabla de hechos central (*fact_cuota_microcredito*) y sus relaciones con las cinco dimensiones definidas.

Figura 8: Modelo dimensional del *datamart* de microcréditos



Nota. Elaboración propia.

2.5. Diseño e implementación del proceso ETL

2.5.1. Extracción desde Cassandra

La extracción de datos desde los microservicios operacionales en Cassandra se implementó mediante funciones AWS Lambda. Actualmente, el proceso ejecuta *cargas completas* diarias (*full load*), donde se extraen todos los registros relevantes de las tablas de usuarios, créditos y cuotas, y se reemplazan completamente las tablas del *datamart* en cada ejecución.

El código soporta un modo incremental que permite extraer únicamente los registros modificados desde la última ejecución (filtrando por campos `created_at` o `updated_at`). Actualmente, esta opción no se encuentra habilitada en el cronograma debido a que el volumen de datos es manejable con cargas completas (aproximadamente 12,500 registros en total entre clientes, créditos y cuotas), y queda disponible como mecanismo de escalabilidad para el futuro.

2.5.2. Transformaciones aplicadas

Las transformaciones se implementan en Python utilizando la librería Pandas y se ejecutan dentro de cada función Lambda después de la extracción. A continuación se describen las principales transformaciones organizadas por tipo:

2.5.2.1. Conversión de tipos de datos

- **Conversión de UUID a cadena:** Los identificadores en Cassandra se almacenan como objetos `uuid.UUID`, los cuales no son compatibles directamente con Redshift. La función auxiliar `convert_uuids_to_strings()` recorre todas las columnas del *DataFrame* y convierte los valores UUID a cadenas de texto.

```
1 def convert_uuids_to_strings(df):
2     for col in df.columns:
3         if df[col].dtype == 'object' and len(df) > 0:
4             first_val = df[col].iloc[0]
5             if first_val is not None and 'uuid.UUID' in str(type(first_val)):
6                 df[col] = df[col].astype(str)
7     return df
```

Listing 2.1: Conversión de UUID a cadena de texto

- **Manejo de fechas nulas:** Los valores NaT (*Not a Time*) generados por Pandas al procesar fechas inválidas se reemplazan por None antes de la carga, evitando errores de inserción en Redshift.
- **Conversión de fechas:** Los campos de fecha se procesan con `pd.to_datetime()` y se formatean como enteros YYYYMMDD para las claves sustitutas de tiempo.

```

1 df['sk_tiempo_vencimiento'] = pd.to_datetime(
2     df['expiration_date'], errors='coerce'
3 ).dt.strftime('%Y%m%d').astype(float).fillna(0).astype(int)

```

Listing 2.2: Cálculo de clave sustituta de tiempo

2.5.2.2. Limpieza y estandarización

- **Limpieza de teléfonos:** En la dimensión cliente, los números telefónicos se limpian eliminando el prefijo internacional +51 y caracteres no numéricos mediante expresiones regulares.

```

1 df['telefono'] = (
2     df['cell_phone']
3     .astype(str)
4     .str.replace(r'^\+51', '', regex=True)
5     .str.replace(r'[s\ -]', '', regex=True)
6 )

```

Listing 2.3: Limpieza de números telefónicos

- **Extracción de número de cuota:** En la tabla de hechos, el número de cuota se extrae del campo description utilizando expresiones regulares.
- **Extracción de días de atraso:** El campo delay contiene valores como “5 días”, de los cuales se extrae el componente numérico.

```

1 df['numero_cuota'] = df['description'].str.extract(
2     r'Cuota (\d+)')
3 ).astype(float).fillna(0).astype(int)
4
5 df['dias_atraso'] = df['delay'].astype(str).str.extract(
6     r'(\d+)')
7 ).astype(float).fillna(0).astype(int)

```

Listing 2.4: Extracción de campos mediante expresiones regulares

2.5.2.3. Integración de datos (merges)

Los datos extraídos de múltiples tablas de Cassandra se integran mediante operaciones merge de Pandas, el siguiente fragmento ilustra la integración de la dimensión cliente:

```
1 # Unir usuarios con empleo
2 df = df_users.merge(
3     df_employment,
4     on='user_id',
5     how='left',
6     suffixes=('', '_emp')
7 )
8
9 # Identificar clientes Waki Planilla
10 payroll_dnis = set(df_payroll['document_number'].astype(str).tolist())
11 df['es_waki_planilla'] = (
12     df['user_origin'].isna() &
13     df['user_origin_app'].isna() &
14     df['dni'].astype(str).isin(payroll_dnis)
15 )
```

Listing 2.5: Integración de datos para dimensión cliente

2.5.2.4. Aplicación de reglas de negocio

Las reglas de negocio se implementan mediante funciones de mapeo que asignan las claves sustitutas correspondientes:

```
1 def map_producto(row):
2     user_origin = row.get('user_origin')
3     user_origin_app = row.get('user_origin_app')
4     bureau_msg = row.get('bureau_msg')
5
6     # Waki Planilla: campos de origen nulos
7     if pd.isna(user_origin) and pd.isna(user_origin_app):
8         if bureau_msg and 'Adelanto sueldo' in str(bureau_msg):
9             return 2 # Adelanto de Sueldo
10            return 1 # Crédito por Convenio
11
12     # Waki Consumo: tiene valor de origen
13     if user_origin == 'Web':
```

```

14     return 3
15     elif user_origin == 'App':
16         return 4
17     # ... otros canales
18     return 7 # Desconocido
19 df['sk_producto'] = df.apply(map_producto, axis=1)

```

Listing 2.6: Mapeo de tipo de producto

2.5.2.5. Cálculo de campos derivados

- **Monto pagado:** Se calcula condicionalmente según el estado del pago.
- **Banderas lógicas:** Se generan indicadores booleanos para facilitar el análisis.

```

1 # Monto pagado condicional
2 df['monto_pagado'] = np.where(
3     (df['payment_state'] == 'Aprobado') | (df['state'] == 'Finalizado'),
4     df['amount'],
5     0
6 )
7
8 # Banderas lógicas
9 df['flag_vencida'] = df['state'] == 'Vencida'
10 df['flag_pagada'] = (df['payment_state'] == 'Aprobado') | (df['state'] == 'Finalizado')
11 df['flag_con_mora'] = df['payment_delay'].fillna(0) > 0

```

Listing 2.7: Cálculo de campos derivados en tabla de hechos

2.5.3. Carga en Amazon Redshift

La carga de datos transformados hacia Amazon Redshift Serverless se realiza mediante el conector `redshift_connector` de Python, que establece una conexión directa al *workgroup* configurado. Las credenciales de acceso se obtienen dinámicamente desde AWS Secrets Manager al inicio de cada ejecución, cabe destacar que el script utilitario `aws_utils.py` extrae estos valores de Secrets Manager y los inyecta en las variables de entorno locales (`os.environ`) instantes antes de establecer la conexión; de esta forma, el código de los clientes de base de datos lee de `os.environ` manteniendo la seguridad y evitando exponer información sensible en el repositorio.

```

1 import redshift_connector
2 def connect(self):
3     self.conn = redshift_connector.connect(
4         host=os.environ.get('REDSHIFT_HOST'),
5         database=os.environ.get('REDSHIFT_DATABASE'),
6         user=os.environ.get('REDSHIFT_USER'),
7         password=os.environ.get('REDSHIFT_PASSWORD'),
8         port=5439
9     )
10    self.cursor = self.conn.cursor()

```

Listing 2.8: Conexión a Redshift Serverless

2.5.3.1. Estrategia de carga completa (Full Load)

La estrategia principal utilizada es la carga completa con reemplazo, implementada en el método `load_dataframe()` del cliente Redshift:

```

1 def load_dataframe(self, df, table_name, schema='dw_microcreditos',
2     if_exists='replace'):
3     full_table = f'{schema}.{table_name}'
4
5     if if_exists == 'replace':
6         self.cursor.execute(f"TRUNCATE TABLE {full_table}")
7
8         # Generar INSERT dinámico
9         columns = ', '.join(df.columns)
10        placeholders = ', '.join(['%s'] * len(df.columns))
11        insert_sql = f"INSERT INTO {full_table} ({columns}) VALUES ({placeholders})"
12
13        # Convertir DataFrame a lista de tuplas
14        data = [tuple(row) for row in df.itertuples(index=False, name=None)]
15
16        # Inserción masiva
17        self.cursor.executemany(insert_sql, data)
18        self.conn.commit()
19
20    return len(df)

```

Listing 2.9: Método `load_dataframe` para carga completa

Esta estrategia garantiza la consistencia de los datos al reemplazar completamente el contenido de cada tabla en cada ejecución, evitando duplicados o registros huérfanos. Las duraciones reales de ejecución de cada función Lambda se documentan más adelante en este capítulo, en la subsección de *Rendimiento del proceso ETL*.

2.5.3.2. Estrategia de carga incremental (Upsert)

Para escenarios donde se requiere actualizar únicamente los registros modificados, el código implementa un patrón de *upsert* mediante tabla de *staging*:

```
1 def upsert_dataframe(self, df, table_name, key_columns,
2     schema='dw_microcreditos'):
3     full_table = f'{schema}.{table_name}'
4     staging_table = f'{schema}.staging_{table_name}'
5
6     # Crear tabla staging temporal
7     self.cursor.execute(f"""
8         CREATE TEMP TABLE {staging_table} (LIKE {full_table})
9     """)
10
11    # Cargar datos en staging
12    self.load_dataframe(df, f'staging_{table_name}', schema,
13        if_exists='append')
14
15    # DELETE de registros existentes que serán actualizados
16    key_conditions = ' AND '.join(
17        [f'{full_table}.{k} = {staging_table}.{k}' for k in key_columns]
18    )
19    self.cursor.execute(f"""
20        DELETE FROM {full_table}
21        USING {staging_table}
22        WHERE {key_conditions}
23    """)
24
25    # INSERT de registros nuevos/actualizados
26    self.cursor.execute(f"""
27        INSERT INTO {full_table}
28        SELECT * FROM {staging_table}
29    """)
30
```

```

31 # Eliminar staging
32 self.cursor.execute(f'DROP TABLE IF EXISTS {staging_table}')
33 self.conn.commit()

```

Listing 2.10: Método upsert_dataframe para carga incremental

Actualmente, las ejecuciones programadas utilizan la estrategia de carga completa debido al volumen reducido de datos, reservando la carga incremental para futuras optimizaciones cuando el volumen de la cartera lo requiera.

2.5.4. Rendimiento del proceso ETL

El rendimiento del proceso ETL se mide a partir de la duración real de ejecución de las seis funciones AWS Lambda que lo componen. Cada función Lambda registra automáticamente en Amazon CloudWatch Logs un evento de tipo REPORT al finalizar su invocación, el cual incluye el tiempo total de ejecución (BilledDuration). Las Figuras 10 a 14 muestran las trazas de ejecución de las seis funciones Lambda durante una corrida diaria en producción.

Figura 9: Tiempo de ejecución de etl-dim-estado-pago en CloudWatch

```

2026-04-10T13:22:13.173-05:00 START RequestId: 0606b10f-2271-4048-bfcb-977094148a72 Version: $LATEST
2026-04-10T13:22:13.174-05:00 [INFO] 2026-04-10T13:22:13.174Z 0606b10f-2271-4048-bfcb-977094148a72 Starting dim_credito ETL
2026-04-10T13:22:13.321-05:00 [INFO] 2026-04-10T13:22:13.321Z 0606b10f-2271-4048-bfcb-977094148a72 Connecting to Cassandra: ['192.168.0.144', '192.168.0.155', '192.168.10.253']
2026-04-10T13:22:13.775-05:00 [INFO] 2026-04-10T13:22:13.774Z 0606b10f-2271-4048-bfcb-977094148a72 Connected to Cassandra successfully
2026-04-10T13:22:13.775-05:00 [INFO] 2026-04-10T13:22:13.775Z 0606b10f-2271-4048-bfcb-977094148a72 Connecting to Redshift: waki-bi-workgroup.894437190166.us-east-1.redshift-serverless.amazonaws.com
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4048-bfcb-977094148a72 Connected to Redshift successfully
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4048-bfcb-977094148a72 Running full load
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4048-bfcb-977094148a72 Extracting credits from microcredit_mag_ms_prod_20191129221827_0915.microcredits
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4048-bfcb-977094148a72 Full mode: all records
2026-04-10T13:22:14.439-05:00 [INFO] 2026-04-10T13:22:14.439Z 0606b10f-2271-4048-bfcb-977094148a72 Extracted 2014 credits
2026-04-10T13:22:14.439-05:00 [INFO] 2026-04-10T13:22:14.439Z 0606b10f-2271-4048-bfcb-977094148a72 Extracting amounts from microcredit_mag_ms_prod_20191129221827_0915.amount
2026-04-10T13:22:14.639-05:00 [INFO] 2026-04-10T13:22:14.639Z 0606b10f-2271-4048-bfcb-977094148a72 Extracted 2029 amount records
2026-04-10T13:22:14.639-05:00 [INFO] 2026-04-10T13:22:14.639Z 0606b10f-2271-4048-bfcb-977094148a72 Extracting disbursements from microcredit_mag_ms_prod_20191129221827_0915.disbursement
2026-04-10T13:22:14.942-05:00 [INFO] 2026-04-10T13:22:14.942Z 0606b10f-2271-4048-bfcb-977094148a72 Extracted 2017 disbursement records
2026-04-10T13:22:14.942-05:00 [INFO] 2026-04-10T13:22:14.942Z 0606b10f-2271-4048-bfcb-977094148a72 Extracting schedules from microcredit_mag_ms_prod_20191129221827_0915.schedule
2026-04-10T13:22:15.118-05:00 [INFO] 2026-04-10T13:22:15.118Z 0606b10f-2271-4048-bfcb-977094148a72 Extracted 2029 schedule records
2026-04-10T13:22:15.118-05:00 [INFO] 2026-04-10T13:22:15.118Z 0606b10f-2271-4048-bfcb-977094148a72 Extracting requests from req_microcredit_ms_prod_20200228233117_0090.microcredits
2026-04-10T13:22:16.420-05:00 [INFO] 2026-04-10T13:22:16.419Z 0606b10f-2271-4048-bfcb-977094148a72 Extracted 5132 request records
2026-04-10T13:22:16.423-05:00 [INFO] 2026-04-10T13:22:16.423Z 0606b10f-2271-4048-bfcb-977094148a72 Transforming 2014 credits with Pandas
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.717Z 0606b10f-2271-4048-bfcb-977094148a72 Transformed 2014 credits
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4048-bfcb-977094148a72 Finalizado: 1558
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4048-bfcb-977094148a72 Retrasado: 261
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4048-bfcb-977094148a72 Activo: 166
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4048-bfcb-977094148a72 Anulado: 26
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4048-bfcb-977094148a72 Cancelado: 2
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4048-bfcb-977094148a72 Pendiente: 1
2026-04-10T13:24:47.556-05:00 [INFO] 2026-04-10T13:24:47.556Z 0606b10f-2271-4048-bfcb-977094148a72 Loaded 2014 rows to dw_microcredits.dim_credito
2026-04-10T13:24:47.558-05:00 [INFO] 2026-04-10T13:24:47.558Z 0606b10f-2271-4048-bfcb-977094148a72 Cassandra connection closed
2026-04-10T13:24:47.558-05:00 [INFO] 2026-04-10T13:24:47.558Z 0606b10f-2271-4048-bfcb-977094148a72 Redshift connection closed
2026-04-10T13:24:47.578-05:00 END RequestId: 0606b10f-2271-4048-bfcb-977094148a72
2026-04-10T13:24:47.578-05:00 REPORT RequestId: 0606b10f-2271-4048-bfcb-977094148a72 Duration: 15440.28 ms Billed Duration: 15600.2 ms Memory Size: 512 MB Max Memory Used: 204 MB Init Duration: 1597.22 ms

```

Nota. Elaboración propia.

Figura 10: Tiempo de ejecución de etl-dim-tiempo en CloudWatch

```
2026-04-10T13:13:41.565-05:00 START RequestId: 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Version: $LATEST
2026-04-10T13:13:41.565-05:00 [INFO] 2026-04-10T13:13:41.565Z 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Starting dim_tiempo ETL
2026-04-10T13:13:41.699-05:00 [INFO] 2026-04-10T13:13:41.699Z 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Generating dim_tiempo from 2019-01-01 to 2026-12-31
2026-04-10T13:13:41.785-05:00 [INFO] 2026-04-10T13:13:41.785Z 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Generated 2922 rows for dim_tiempo
2026-04-10T13:13:41.786-05:00 [INFO] 2026-04-10T13:13:41.786Z 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Connecting to Redshift: waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com
2026-04-10T13:13:42.835-05:00 [INFO] 2026-04-10T13:13:42.835Z 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Connected to Redshift successfully
2026-04-10T13:17:28.623-05:00 [INFO] 2026-04-10T13:17:28.623Z 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Loaded 2922 rows to dw_microcreditos.dim_tiempo
2026-04-10T13:17:28.624-05:00 [INFO] 2026-04-10T13:17:28.624Z 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Redshift connection closed
2026-04-10T13:17:28.626-05:00 END RequestId: 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb
2026-04-10T13:17:28.626-05:00 REPORT RequestId: 9e0e250c-34d3-4738-82cc-c6acfa0a2dfb Duration: 227068.96 ms Billed Duration: 228342 ms Memory Size: 512 MB Max Memory Used: 167 MB Init Duration: 1288.89 ms
No newer events at this moment. Auto retry paused. Resume
```

Nota. Elaboración propia.

Figura 11: Tiempo de ejecución de etl-dim-producto en CloudWatch

```
2026-04-10T13:17:37.672-05:00 START RequestId: 8362deea-e7a7-455b-a854-8abfec613251 Version: $LATEST
2026-04-10T13:17:37.673-05:00 [INFO] 2026-04-10T13:17:37.673Z 8362deea-e7a7-455b-a854-8abfec613251 Starting dim_producto ETL
2026-04-10T13:17:37.778-05:00 [INFO] 2026-04-10T13:17:37.778Z 8362deea-e7a7-455b-a854-8abfec613251 Generating dim_producto
2026-04-10T13:17:37.780-05:00 [INFO] 2026-04-10T13:17:37.780Z 8362deea-e7a7-455b-a854-8abfec613251 Generated 7 rows for dim_producto
2026-04-10T13:17:37.780-05:00 [INFO] 2026-04-10T13:17:37.780Z 8362deea-e7a7-455b-a854-8abfec613251 Connecting to Redshift: waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com
2026-04-10T13:17:38.025-05:00 [INFO] 2026-04-10T13:17:38.025Z 8362deea-e7a7-455b-a854-8abfec613251 Connected to Redshift successfully
2026-04-10T13:17:40.495-05:00 [INFO] 2026-04-10T13:17:40.495Z 8362deea-e7a7-455b-a854-8abfec613251 Loaded 7 rows to dw_microcreditos.dim_producto
2026-04-10T13:17:40.495-05:00 [INFO] 2026-04-10T13:17:40.495Z 8362deea-e7a7-455b-a854-8abfec613251 Redshift connection closed
2026-04-10T13:17:40.497-05:00 END RequestId: 8362deea-e7a7-455b-a854-8abfec613251
2026-04-10T13:17:40.497-05:00 REPORT RequestId: 8362deea-e7a7-455b-a854-8abfec613251 Duration: 2824.69 ms Billed Duration: 4328 ms Memory Size: 512 MB Max Memory Used: 165 MB Init Duration: 1302.77 ms
```

Nota. Elaboración propia.

Figura 12: Tiempo de ejecución de etl-dim-cliente en CloudWatch

```
2026-04-10T13:17:43.254-05:00 START RequestId: 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Version: $LATEST
2026-04-10T13:17:43.255-05:00 [INFO] 2026-04-10T13:17:43.255Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Starting dim_cliente ETL
2026-04-10T13:17:43.390-05:00 [INFO] 2026-04-10T13:17:43.390Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Connecting to Cassandra: ["192.168.0.144", "192.168.6.155", "192.168.10.253"]
2026-04-10T13:17:46.791-05:00 [INFO] 2026-04-10T13:17:46.791Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Connected to Cassandra successfully
2026-04-10T13:17:46.791-05:00 [INFO] 2026-04-10T13:17:46.791Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Connecting to Redshift: waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com
2026-04-10T13:17:47.152-05:00 [INFO] 2026-04-10T13:17:47.152Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Connected to Redshift successfully
2026-04-10T13:17:47.152-05:00 [INFO] 2026-04-10T13:17:47.152Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Running full load
2026-04-10T13:17:47.719Z [INFO] 2026-04-10T13:17:47.719Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Extracting users from user_management_ms_prod_20191129175655_0139_user
2026-04-10T13:17:47.152-05:00 [INFO] 2026-04-10T13:17:47.152Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Full mode: all records
2026-04-10T13:17:47.720-05:00 [INFO] 2026-04-10T13:17:47.720Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Extracted 3018 users
2026-04-10T13:17:47.720-05:00 [INFO] 2026-04-10T13:17:47.720Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Extracting employment from user_management_ms_prod_20191129175655_0139_user_employment
2026-04-10T13:17:47.900-05:00 [INFO] 2026-04-10T13:17:47.900Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Extracted 3051 employment records
2026-04-10T13:17:47.900-05:00 [INFO] 2026-04-10T13:17:47.900Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Extracting payroll from admission_ms_prod_20190518184225_0799_payroll
2026-04-10T13:17:48.382Z [INFO] 2026-04-10T13:17:48.382Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Extracted 5315 payroll records
2026-04-10T13:17:48.402-05:00 [INFO] 2026-04-10T13:17:48.402Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Transforming 3018 users with Pandas
2026-04-10T13:17:48.586-05:00 [INFO] 2026-04-10T13:17:48.586Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Transformed 3018 clients
2026-04-10T13:17:48.586-05:00 [INFO] 2026-04-10T13:17:48.586Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Waki Planilla: 964
2026-04-10T13:17:48.586-05:00 [INFO] 2026-04-10T13:17:48.586Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Waki Consumo: 2046
2026-04-10T13:22:09.126-05:00 [INFO] 2026-04-10T13:22:09.126Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Loaded 3018 rows to dw_microcreditos.dim_cliente
2026-04-10T13:22:09.129-05:00 [INFO] 2026-04-10T13:22:09.129Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Cassandra connection closed
2026-04-10T13:22:09.129-05:00 [INFO] 2026-04-10T13:22:09.129Z 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Redshift connection closed
2026-04-10T13:22:09.140-05:00 END RequestId: 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40
2026-04-10T13:22:09.141-05:00 REPORT RequestId: 7e9fa630-a2fc-4f2b-9433-1bfc66fbab40 Duration: 265885.64 ms Billed Duration: 267521 ms Memory Size: 512 MB Max Memory Used: 196 MB Init Duration: 1634.51 ms
```

Nota. Elaboración propia.

Figura 13: Tiempo de ejecución de etl-dim-credito en CloudWatch

```

2026-04-10T13:22:13.173-05:00 START RequestId: 0606b10f-2271-4040-bfcb-977094148a72 Version: $LATEST
2026-04-10T13:22:13.174-05:00 [INFO] 2026-04-10T13:22:13.174Z 0606b10f-2271-4040-bfcb-977094148a72 Starting dim_credito ETL
2026-04-10T13:22:13.321-05:00 [INFO] 2026-04-10T13:22:13.321Z 0606b10f-2271-4040-bfcb-977094148a72 Connecting to Cassandra: ["192.168.0.144", "192.168.6.155", "192.168.10.253"]
2026-04-10T13:22:13.775-05:00 [INFO] 2026-04-10T13:22:13.774Z 0606b10f-2271-4040-bfcb-977094148a72 Connected to Cassandra successfully
2026-04-10T13:22:13.775-05:00 [INFO] 2026-04-10T13:22:13.775Z 0606b10f-2271-4040-bfcb-977094148a72 Connecting to Redshift: wsl-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4040-bfcb-977094148a72 Connected to Redshift successfully
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4040-bfcb-977094148a72 Running full load
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4040-bfcb-977094148a72 Extracting credits from microcredit_mag_ms_prod_20191129221827_0915.microcredits
2026-04-10T13:22:14.034-05:00 [INFO] 2026-04-10T13:22:14.034Z 0606b10f-2271-4040-bfcb-977094148a72 Full mode: all records
2026-04-10T13:22:14.439-05:00 [INFO] 2026-04-10T13:22:14.439Z 0606b10f-2271-4040-bfcb-977094148a72 Extracted 2014 credits
2026-04-10T13:22:14.439-05:00 [INFO] 2026-04-10T13:22:14.439Z 0606b10f-2271-4040-bfcb-977094148a72 Extracting amounts from microcredit_mag_ms_prod_20191129221827_0915.amount
2026-04-10T13:22:14.639-05:00 [INFO] 2026-04-10T13:22:14.639Z 0606b10f-2271-4040-bfcb-977094148a72 Extracted 2029 amount records
2026-04-10T13:22:14.639-05:00 [INFO] 2026-04-10T13:22:14.639Z 0606b10f-2271-4040-bfcb-977094148a72 Extracting disbursements from microcredit_mag_ms_prod_20191129221827_0915.disbursement
2026-04-10T13:22:14.942-05:00 [INFO] 2026-04-10T13:22:14.942Z 0606b10f-2271-4040-bfcb-977094148a72 Extracted 2017 disbursement records
2026-04-10T13:22:14.942-05:00 [INFO] 2026-04-10T13:22:14.942Z 0606b10f-2271-4040-bfcb-977094148a72 Extracting schedules from microcredit_mag_ms_prod_20191129221827_0915.schedule
2026-04-10T13:22:15.118-05:00 [INFO] 2026-04-10T13:22:15.118Z 0606b10f-2271-4040-bfcb-977094148a72 Extracted 2029 schedule records
2026-04-10T13:22:15.118-05:00 [INFO] 2026-04-10T13:22:15.118Z 0606b10f-2271-4040-bfcb-977094148a72 Extracting requests from req_microcredit_tms_prod_2020022823117_0090.microcredits
2026-04-10T13:22:16.420-05:00 [INFO] 2026-04-10T13:22:16.419Z 0606b10f-2271-4040-bfcb-977094148a72 Extracted 5132 request records
2026-04-10T13:22:16.423-05:00 [INFO] 2026-04-10T13:22:16.423Z 0606b10f-2271-4040-bfcb-977094148a72 Transforming 2014 credits with Pandas
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.717Z 0606b10f-2271-4040-bfcb-977094148a72 Transformed 2014 credits
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4040-bfcb-977094148a72 Finalizando: 1558
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4040-bfcb-977094148a72 Retirando: 261
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4040-bfcb-977094148a72 Activo: 186
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4040-bfcb-977094148a72 Anulado: 26
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4040-bfcb-977094148a72 Cancelado: 2
2026-04-10T13:22:16.718-05:00 [INFO] 2026-04-10T13:22:16.718Z 0606b10f-2271-4040-bfcb-977094148a72 Pendiente: 1
2026-04-10T13:24:47.556-05:00 [INFO] 2026-04-10T13:24:47.556Z 0606b10f-2271-4040-bfcb-977094148a72 Loaded 2014 rows to dw_microcredit_dim_credito
2026-04-10T13:24:47.558-05:00 [INFO] 2026-04-10T13:24:47.558Z 0606b10f-2271-4040-bfcb-977094148a72 Cassandra connection closed
2026-04-10T13:24:47.558-05:00 [INFO] 2026-04-10T13:24:47.558Z 0606b10f-2271-4040-bfcb-977094148a72 Redshift connection closed
2026-04-10T13:24:47.578-05:00 END RequestId: 0606b10f-2271-4040-bfcb-977094148a72
2026-04-10T13:24:47.578-05:00 REPORT RequestId: 0606b10f-2271-4040-bfcb-977094148a72 Duration: 15448M.28 ns Billed Duration: 15600Z ns Memory Size: 512 MB Max Memory Used: 204 MB Init Duration: 1597.22 ms
    
```

Nota. Elaboración propia.

Figura 14: Tiempo de ejecución de etl-fact-cuota en CloudWatch

```

2026-04-10T13:24:50.000-05:00 START RequestId: 0508a255-3a03-43a2-c081-6eb0a0b0e042 Version: $LATEST
2026-04-10T13:24:50.012-05:00 [INFO] 2026-04-10T13:24:50.012Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Starting fact_cuota_microcredit ETL
2026-04-10T13:24:50.127-05:00 [INFO] 2026-04-10T13:24:50.127Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Connecting to Cassandra: ["192.168.0.144", "192.168.6.155", "192.168.10.253"]
2026-04-10T13:24:50.413-05:00 [INFO] 2026-04-10T13:24:50.413Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Connected to Cassandra successfully
2026-04-10T13:24:50.413-05:00 [INFO] 2026-04-10T13:24:50.413Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Connecting to Redshift: wsl-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com
2026-04-10T13:24:50.661-05:00 [INFO] 2026-04-10T13:24:50.661Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Connected to Redshift successfully
2026-04-10T13:24:50.661-05:00 [INFO] 2026-04-10T13:24:50.661Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Running Full load
2026-04-10T13:24:50.661-05:00 [INFO] 2026-04-10T13:24:50.661Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracting quotas from microcredit_mag_ms_prod_20191129221827_0915.payment.schedule
2026-04-10T13:24:50.661-05:00 [INFO] 2026-04-10T13:24:50.661Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Full mode: all records
2026-04-10T13:24:50.977-05:00 [WARNING] 2026-04-10T13:24:50.977Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Server warning: Read 5000 live rows and 2687 hotstore cells for query SELECT * FROM microcredit_tms_prod_20191129221827_0915.payment.schedule LIMIT 5000 ALLOW FILTERING.
2026-04-10T13:24:51.252-05:00 [INFO] 2026-04-10T13:24:51.252Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracted 7391 quotas
2026-04-10T13:24:51.750-05:00 [INFO] 2026-04-10T13:24:51.750Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracting quota details from microcredit_mag_ms_prod_20191129221827_0915.quota_detail
2026-04-10T13:24:51.474-05:00 [INFO] 2026-04-10T13:24:51.474Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracted 7418 quota detail records
2026-04-10T13:24:51.674-05:00 [INFO] 2026-04-10T13:24:51.674Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracting schedules from microcredit_mag_ms_prod_20191129221827_0915.schedule
2026-04-10T13:24:52.640-05:00 [INFO] 2026-04-10T13:24:52.640Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracted 2029 schedule records
2026-04-10T13:24:52.640-05:00 [INFO] 2026-04-10T13:24:52.640Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracting microcredits from microcredit_mag_ms_prod_20191129221827_0915.microcredits
2026-04-10T13:24:52.795-05:00 [INFO] 2026-04-10T13:24:52.795Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracted 2054 microcredit records
2026-04-10T13:24:52.795-05:00 [INFO] 2026-04-10T13:24:52.795Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracting requests from req_microcredit_tms_prod_2020022823117_0090.microcredits
2026-04-10T13:24:53.319-05:00 [INFO] 2026-04-10T13:24:53.319Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Extracted 5132 request records
2026-04-10T13:24:53.319-05:00 [INFO] 2026-04-10T13:24:53.319Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Transforming 7391 quotas with Pandas
2026-04-10T13:24:53.762-05:00 [INFO] 2026-04-10T13:24:53.762Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Transformed 7391 quotas
2026-04-10T13:24:53.762-05:00 [INFO] 2026-04-10T13:24:53.762Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Pagados: 5599
2026-04-10T13:24:53.762-05:00 [INFO] 2026-04-10T13:24:53.762Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Vencidos: 471
2026-04-10T13:24:53.762-05:00 [INFO] 2026-04-10T13:24:53.762Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Sin moro: 3590
2026-04-10T13:24:53.441-05:00 [INFO] 2026-04-10T13:24:53.441Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Loaded 7391 rows to dw_microcredit_fact_cuota
2026-04-10T13:24:56.446-05:00 [INFO] 2026-04-10T13:24:56.446Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Cassandra connection closed
2026-04-10T13:24:56.446-05:00 [INFO] 2026-04-10T13:24:56.446Z 0508a255-3a03-43a2-c081-6eb0a0b0e042 Redshift connection closed
2026-04-10T13:24:56.467-05:00 END RequestId: 0508a255-3a03-43a2-c081-6eb0a0b0e042
    
```

Nota. Elaboración propia.

La Tabla 17 consolida las duraciones registradas en cada función, obtenidas a partir del campo BilledDuration reportado por AWS Lambda al finalizar cada invocación.

Tabla 17: Duración de ejecución de las funciones Lambda del ETL

Función Lambda	Duración (ms)	Duración (s)
etl-dim-tiempo	2 779	2.78
etl-dim-estado-pago	2 824	2.82
etl-dim-producto	2 824	2.82
etl-dim-cliente	26 752	26.75
etl-dim-credito	15 604	15.60
etl-fact-cuota	607 135	607.13
Total	657 918	~10 min 58 s

Nota. Elaboración propia.

Los tiempos registrados confirman que el proceso ETL completo se ejecuta en aproximadamente once minutos. La función etl-fact-cuota concentra cerca del 92 % del tiempo total de ejecución, debido a que procesa el mayor volumen de registros del datamart (la tabla de hechos a nivel de cuota) y aplica las transformaciones más complejas, incluyendo múltiples operaciones de *merge* y mapeo de claves sustitutas hacia las cinco dimensiones. Estos valores constituyen una medida de eficiencia interna de la solución construida; la experiencia de tiempo del usuario final al consumir los reportes se aborda en el Capítulo 3.

2.5.5. Aseguramiento de calidad del ETL

Para garantizar que las transformaciones aplicadas por el proceso ETL sobre los datos extraídos de Cassandra producen los valores esperados, se implementaron 37 pruebas unitarias con *pytest* que validan la lógica de transformación de cada componente del ETL. Las pruebas se ejecutan de forma aislada, sin conexión a Cassandra ni Redshift, utilizando datos de prueba construidos localmente para garantizar que cada función de transformación produzca los resultados esperados de manera reproducible. La Tabla 18 resume la cobertura por componente.

Tabla 18: Cobertura de pruebas unitarias por componente del ETL

Componente	Pruebas
Conversión de identificadores UUID	4
Mapeo de tipos de producto	7
Mapeo de estados de pago	8
Extracción de campos mediante regex	2
Cálculo de campos derivados y banderas	5
Limpieza de números telefónicos	3
Generación de la dimensión tiempo	8
Total	37

Nota. Elaboración propia.

La Figura 15 muestra el resultado de la ejecución completa de la suite de pruebas con *pytest*, confirmando que las 37 pruebas pasan exitosamente. El código fuente completo de las pruebas se presenta en el Anexo D.

Figura 15: Resultado de ejecución de las 37 pruebas unitarias del ETL

```
(!venv) gonzalourrutiaquequezana@MacBook-Air-de-Gonzalo waki-etl % pytest tests/test_transformations.py -v
platform darwin -- Python 3.12.11, pytest-9.0.3, pluggy-1.6.0 -- /Users/gonzalourrutiaquequezana/Documents/etl-tesis/waki-etl/venv/bin/python3.12
cachedir: .pytest_cache
rootdir: /Users/gonzalourrutiaquequezana/Documents/etl-tesis/waki-etl
collected 37 items

tests/test_transformations.py::TestConvertUuidsToStrings::test_convierte_uuids_a_strings PASSED [ 2%]
tests/test_transformations.py::TestConvertUuidsToStrings::test_dataframe_vacio_retorna_vacio PASSED [ 5%]
tests/test_transformations.py::TestConvertUuidsToStrings::test_none_retorna_none PASSED [ 8%]
tests/test_transformations.py::TestConvertUuidsToStrings::test_columnas_sin_uuid_no_cambian PASSED [ 10%]
tests/test_transformations.py::TestMapProducto::test_waki_planilla_credito_convenio PASSED [ 13%]
tests/test_transformations.py::TestMapProducto::test_waki_planilla_adelanto_sueldo PASSED [ 16%]
tests/test_transformations.py::TestMapProducto::test_waki_consumo_web PASSED [ 18%]
tests/test_transformations.py::TestMapProducto::test_waki_consumo_app PASSED [ 21%]
tests/test_transformations.py::TestMapProducto::test_waki_consumo_simulador PASSED [ 24%]
tests/test_transformations.py::TestMapProducto::test_waki_consumo_facebook PASSED [ 27%]
tests/test_transformations.py::TestMapProducto::test_origen_desconocido_default_web PASSED [ 29%]
tests/test_transformations.py::TestMapEstadoPago::test_activo_pendiente PASSED [ 32%]
tests/test_transformations.py::TestMapEstadoPago::test_pendiente_pendiente PASSED [ 35%]
tests/test_transformations.py::TestMapEstadoPago::test_vencida_pendiente PASSED [ 37%]
tests/test_transformations.py::TestMapEstadoPago::test_finalizado_aprobado PASSED [ 40%]
tests/test_transformations.py::TestMapEstadoPago::test_vencida_aprobado PASSED [ 43%]
tests/test_transformations.py::TestMapEstadoPago::test_pago_rechazado PASSED [ 45%]
tests/test_transformations.py::TestMapEstadoPago::test_refinanciado PASSED [ 48%]
tests/test_transformations.py::TestMapEstadoPago::test_castigado PASSED [ 51%]
tests/test_transformations.py::TestExtraccionRegex::test_extraer_numero_cuota PASSED [ 54%]
tests/test_transformations.py::TestExtraccionRegex::test_extraer_dias_atraso PASSED [ 56%]
tests/test_transformations.py::TestCamposDerivados::test_monto_pagado_aprobado PASSED [ 59%]
tests/test_transformations.py::TestCamposDerivados::test_flag_vencida PASSED [ 62%]
tests/test_transformations.py::TestCamposDerivados::test_flag_pagada PASSED [ 64%]
tests/test_transformations.py::TestCamposDerivados::test_flag_con_mora PASSED [ 67%]
tests/test_transformations.py::TestCamposDerivados::test_sk_tiempo_vencimiento PASSED [ 70%]
tests/test_transformations.py::TestLimpiezaTelefonos::test_elimina_prefijo_51 PASSED [ 72%]
tests/test_transformations.py::TestLimpiezaTelefonos::test_elimina_espacios_y_guiones PASSED [ 75%]
tests/test_transformations.py::TestLimpiezaTelefonos::test_numero_limpio_no_cambia PASSED [ 78%]
tests/test_transformations.py::TestGeneratedimTiempo::test_genera_cantidad_correcta_de_filas PASSED [ 81%]
tests/test_transformations.py::TestGeneratedimTiempo::test_sk_tiempo_formato_YYYYMMDD PASSED [ 83%]
tests/test_transformations.py::TestGeneratedimTiempo::test_nombre_mes_en_español PASSED [ 86%]
tests/test_transformations.py::TestGeneratedimTiempo::test_trimestre_correcto PASSED [ 89%]
tests/test_transformations.py::TestGeneratedimTiempo::test_fin_de_semana PASSED [ 91%]
tests/test_transformations.py::TestGeneratedimTiempo::test_dia_laboral PASSED [ 94%]
tests/test_transformations.py::TestGeneratedimTiempo::test_un_solo_dia PASSED [ 97%]
tests/test_transformations.py::TestGeneratedimTiempo::test_anio_bisiesto PASSED [100%]

===== 37 passed in 0.33s =====
(!venv) gonzalourrutiaquequezana@MacBook-Air-de-Gonzalo waki-etl %
```

Nota. Elaboración propia.

Esto evidencia que la lógica de transformación aplicada al cargar los datos hacia el Data Warehouse produce los resultados esperados, sosteniendo junto con la reconciliación de volúmenes (Capítulo 3) la exactitud de los datos disponibles para el análisis.

2.6. Diseño e implementación de dashboards en Power BI

2.6.1. Conexión con el Data Warehouse

La conexión entre Power BI y Amazon Redshift Serverless requirió el diseño de una arquitectura híbrida debido a que el *workgroup* de Redshift se encuentra desplegado dentro de una VPC privada, sin endpoint público accesible desde internet. Esta configuración responde a requisitos de seguridad de la empresa, que establece que las bases de datos no deben exponerse directamente a conexiones externas, para resolver este desafío, se implementó una solución basada en una instancia EC2 Windows que actúa como puente entre Power BI y Redshift, soportando dos flujos de trabajo diferenciados: uno para el desarrollo local de reportes y otro para la actualización automática de dashboards publicados en la nube.

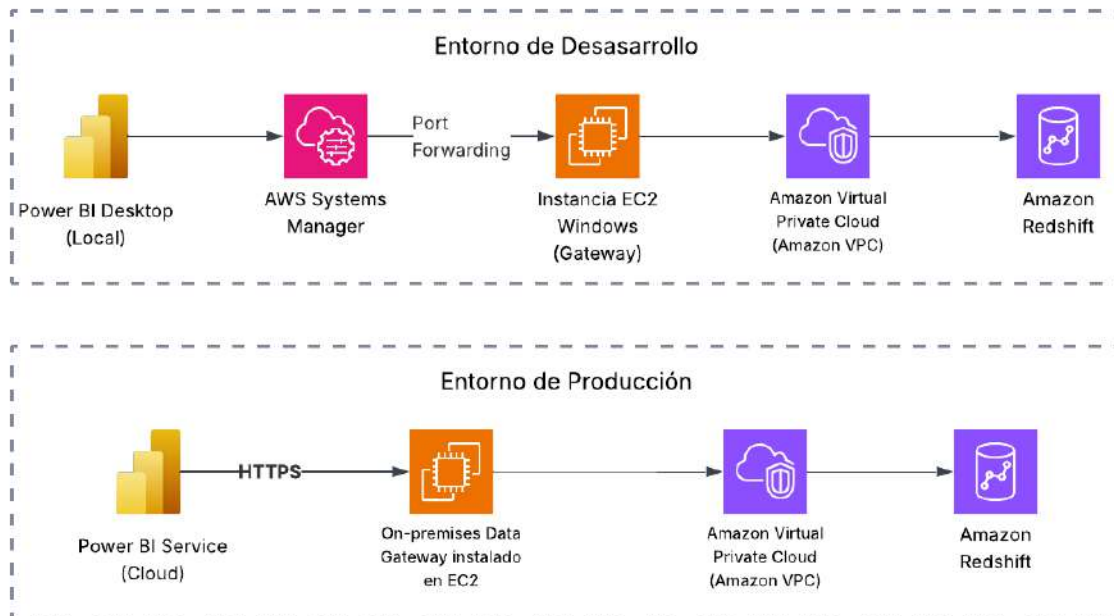
2.6.1.1. Arquitectura de conexión híbrida

La solución implementada integra los siguientes componentes:

- **Amazon Redshift Serverless:** Data warehouse alojado en una VPC privada, accesible únicamente desde recursos dentro de la misma red virtual o a través de conexiones autorizadas explícitamente mediante Security Groups.
- **Instancia EC2 Windows (Gateway/Bastión):** Servidor intermedio ubicado en la misma VPC que Redshift, configurado con acceso autorizado al puerto 5439 del *workgroup*, esta instancia cumple dos funciones: actúa como destino de túneles SSM desde estaciones de trabajo locales y hospeda el *On-premises Data Gateway* de Microsoft para la integración con Power BI Service.
- **AWS Systems Manager (SSM) Port Forwarding:** Servicio que permite establecer túneles TCP encriptados desde una máquina local hacia recursos dentro de la VPC sin necesidad de configurar VPN o abrir puertos públicos en grupos de seguridad; el túnel se establece bajo demanda y se autentica mediante credenciales IAM del desarrollador.
- **On-premises Data Gateway:** Servicio de Microsoft instalado en la instancia EC2 que actúa como proxy entre Power BI Service (nube) y Redshift (VPC privada), el Gateway se registra en el tenant de Power BI de la organización y gestiona las consultas desde los dashboards publicados, ejecutándolas contra Redshift y devolviendo los resultados a través de una conexión HTTPS saliente hacia los servicios de Microsoft.

La figura 16 ilustra la arquitectura híbrida implementada, mostrando los dos flujos de trabajo diferenciados para el entorno de desarrollo y producción.

Figura 16: Arquitectura híbrida de conexión entre Power BI y Amazon Redshift Serverless



Nota. Elaboración propia.

2.6.1.2. Flujo de trabajo en entorno de desarrollo

Durante la fase de desarrollo y diseño de reportes, el analista de BI trabaja desde una estación local (Power BI Desktop) y requiere acceso directo al Data Warehouse para realizar consultas exploratorias, construir visualizaciones y validar modelos de datos; dado que Redshift no es accesible públicamente, el proceso de conexión involucra los siguientes pasos:

1. **Inicio de la infraestructura:** La instancia EC2 Gateway se enciende desde la consola de AWS únicamente durante las sesiones de trabajo, optimizando costos operativos, al encenderse, la instancia obtiene una IP privada dentro de la VPC y queda lista para recibir sesiones SSM.
2. **Establecimiento del túnel SSM:** Utilizando AWS CLI desde la estación de trabajo, se ejecuta el comando `aws ssm start-session` con el documento `AWS-StartPortForwardingSession-ToRemoteHost`, que crea un canal encriptado desde el puerto local 5439 de la máquina del desarrollador hasta el endpoint de Redshift (puerto 5439), atravesando la instancia EC2 como nodo intermedio, este túnel no requiere configuración de VPN ni apertura de puertos SSH públicos, y se autentica mediante las credenciales IAM del usuario.

3. **Configuración del puente de red:** En entornos donde Power BI Desktop se ejecuta en una máquina virtual (por ejemplo, Windows sobre Parallels en macOS), el túnel SSM creado en el host no es directamente accesible desde la VM; para resolver esto, se configura un bridge TCP adicional utilizando socat, que escucha en la interfaz de red compartida (puerto 5440) y redirige el tráfico al túnel local (puerto 5439), esto permite que la máquina virtual acceda al túnel como si fuera un servidor remoto en la red.
4. **Resolución DNS local:** El conector de Redshift en Power BI valida que el hostname del servidor coincida con el certificado SSL presentado por Redshift. Para permitir la conexión a través del puerto redirigido (5440), se añade una entrada en el archivo hosts de Windows que resuelve el FQDN del *workgroup* de Redshift hacia la IP del host (gateway de Parallels), evitando errores de validación SSL sin comprometer la seguridad de la conexión.
5. **Conexión desde Power BI Desktop:** Con la infraestructura de túnel activa, Power BI Desktop se conecta a Redshift especificando el FQDN completo del *workgroup* con el puerto redirigido (5440), la base de datos prod y credenciales de usuario estándar. Power BI establece la conexión a través del túnel como si estuviera accediendo directamente a Redshift, permitiendo trabajar en modo *DirectQuery* o importar datos según las necesidades del análisis.

Este flujo permite desarrollar reportes de manera ágil sin necesidad de exponer Redshift públicamente ni configurar infraestructura de red compleja, delegando la seguridad a IAM y SSM.

2.6.1.3. Flujo de trabajo en entorno de producción

Una vez que los reportes han sido diseñados y validados, se publican en Power BI Service para que los usuarios finales (administradora, analista de créditos, junta de socios) puedan acceder a los dashboards desde cualquier dispositivo con conexión a internet, sin embargo, Power BI Service (plataforma SaaS en la nube de Microsoft) no puede conectarse directamente a Redshift debido a que este se encuentra en una VPC privada. Para habilitar la actualización automática de datos en los dashboards publicados, se configuró el *On-premises Data Gateway* en la instancia EC2:

1. **Instalación y registro del Gateway:** El *On-premises Data Gateway* se instala como un servicio de Windows en la instancia EC2, ejecutándose en modo estándar (*Standard Mode*). Durante la instalación, el Gateway se registra en el tenant de Power BI de la organización mediante autenticación de Azure Active Directory, quedando visible en la consola de administración de gateways en Power BI Service con un nombre identificador (por ejemplo, Gateway-Redshift-Prod).

2. **Configuración de la fuente de datos:** En Power BI Service, se crea una nueva conexión de tipo *On-premises* vinculada al Gateway registrado. Se especifica el endpoint original de Redshift (con puerto 5439), la base de datos prod y las credenciales de conexión, que el Gateway almacena de forma encriptada. A partir de este momento, todos los reportes publicados que utilicen esta fuente de datos ejecutarán sus consultas a través del Gateway.
3. **Autorización de red en AWS:** Para que el Gateway pueda comunicarse con Redshift, se añade una regla de entrada (*inbound rule*) en el Security Group asociado al *workgroup* de Redshift, autorizando el tráfico TCP en el puerto 5439 desde la IP privada de la instancia EC2 Gateway, esto garantiza que solo el Gateway puede conectarse a Redshift desde dentro de la VPC, sin abrir acceso a otros recursos.
4. **Actualización programada:** Una vez configurado el Gateway, los dashboards publicados en Power BI Service pueden programar actualizaciones automáticas de datos (por ejemplo, diarias a las 8:00 AM). Durante cada actualización, Power BI Service envía la solicitud al Gateway a través de un canal HTTPS saliente, el Gateway ejecuta las consultas contra Redshift desde la VPC, obtiene los resultados y los devuelve encriptados a Power BI Service, donde se actualizan los conjuntos de datos y visualizaciones.

Este enfoque permite que los usuarios finales accedan a información actualizada sin necesidad de intervención manual, manteniendo la seguridad del Data Warehouse y cumpliendo con los estándares de arquitectura en la nube de la empresa.

2.6.1.4. Consideraciones de seguridad y optimización de costos

La arquitectura de conexión híbrida implementada presenta las siguientes ventajas desde el punto de vista de seguridad y costos:

- **Sin exposición pública:** Redshift no tiene endpoint público ni permite conexiones directas desde internet, reduciendo la superficie de ataque y cumpliendo con las políticas de seguridad de Holding Waki SAC (Administración de Holding Waki S.A.C., 2025).
- **Autenticación centralizada:** Las conexiones desde el entorno de desarrollo se autentican mediante AWS IAM (sin credenciales embebidas en configuraciones locales), mientras que el Gateway utiliza credenciales encriptadas administradas por Microsoft.
- **Uso optimizado de recursos:** La instancia EC2 Gateway se enciende únicamente durante las sesiones de desarrollo o cuando se requiere actualización de reportes, y se apaga automáticamente fuera del horario laboral, optimizando costos de computación en AWS.
- **Trazabilidad:** Todos los accesos a Redshift desde el Gateway quedan registrados en los logs de CloudWatch, permitiendo identificar el origen de las consultas ejecutadas.

2.6.2. Estructura y diseño de los dashboards

El sistema de dashboards se implementó como un único archivo Power BI (.pbix) que contiene cuatro páginas organizadas en dos niveles de análisis: gerencial y operativo. La tabla 19 presenta la estructura completa del sistema.

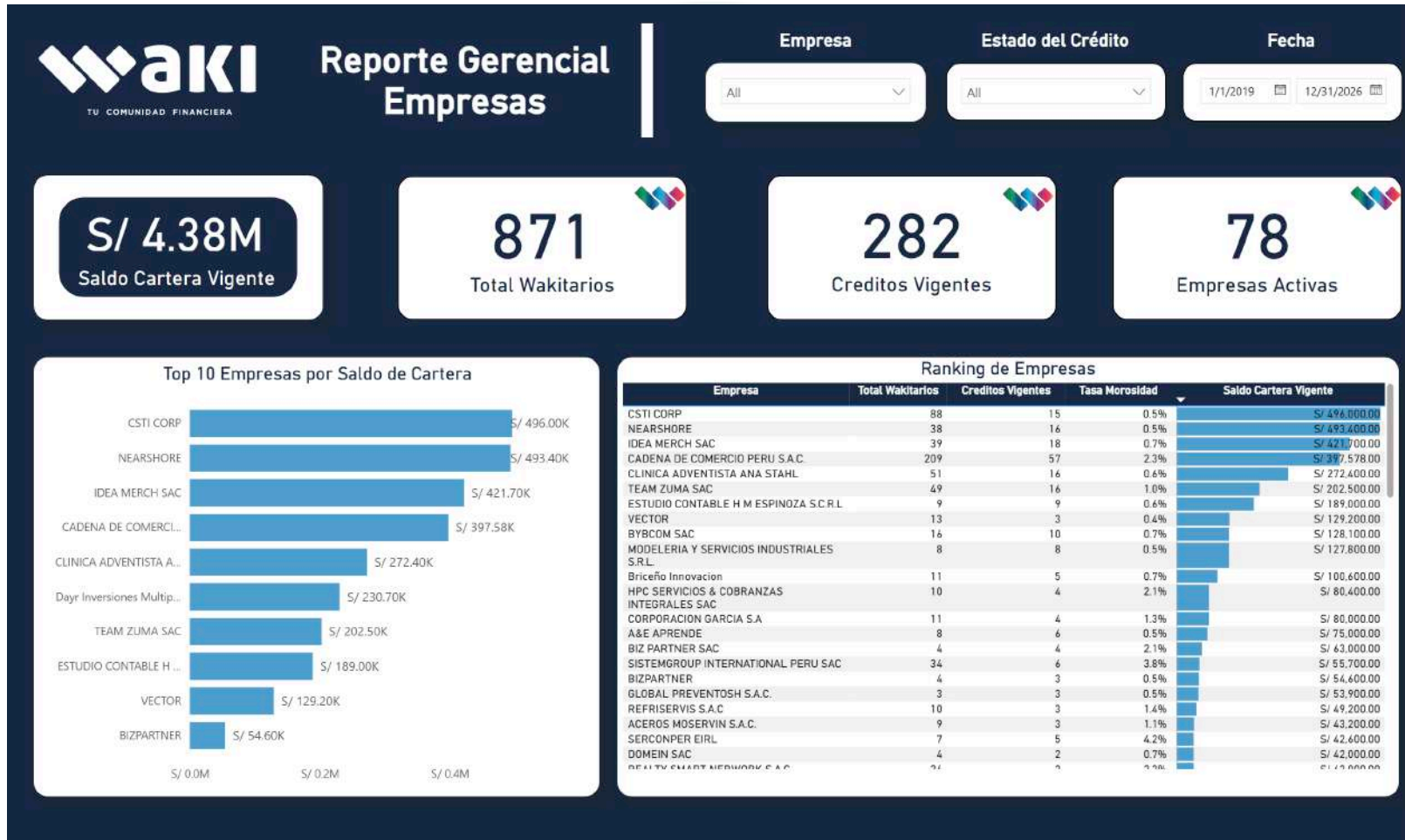
Tabla 19: Estructura del sistema de dashboards

Pág.	Nombre	Tipo	Usuario Destino
1	Reporte Gerencial - Empresas	Gerencial	Junta de Socios
2	Reporte Gerencial - Empresas Detalle	Gerencial	Junta de Socios
3	Reporte Operativo - Cobranza	Operativo	Administradora, Analista
4	Reporte Operativo - Alertas y Riesgo	Operativo	Administradora, Analista

Nota. Elaboración propia.

2.6.2.1. Página 1: Reporte Gerencial - Empresas

Figura 17: Dashboard Gerencial - Vista General de Empresas

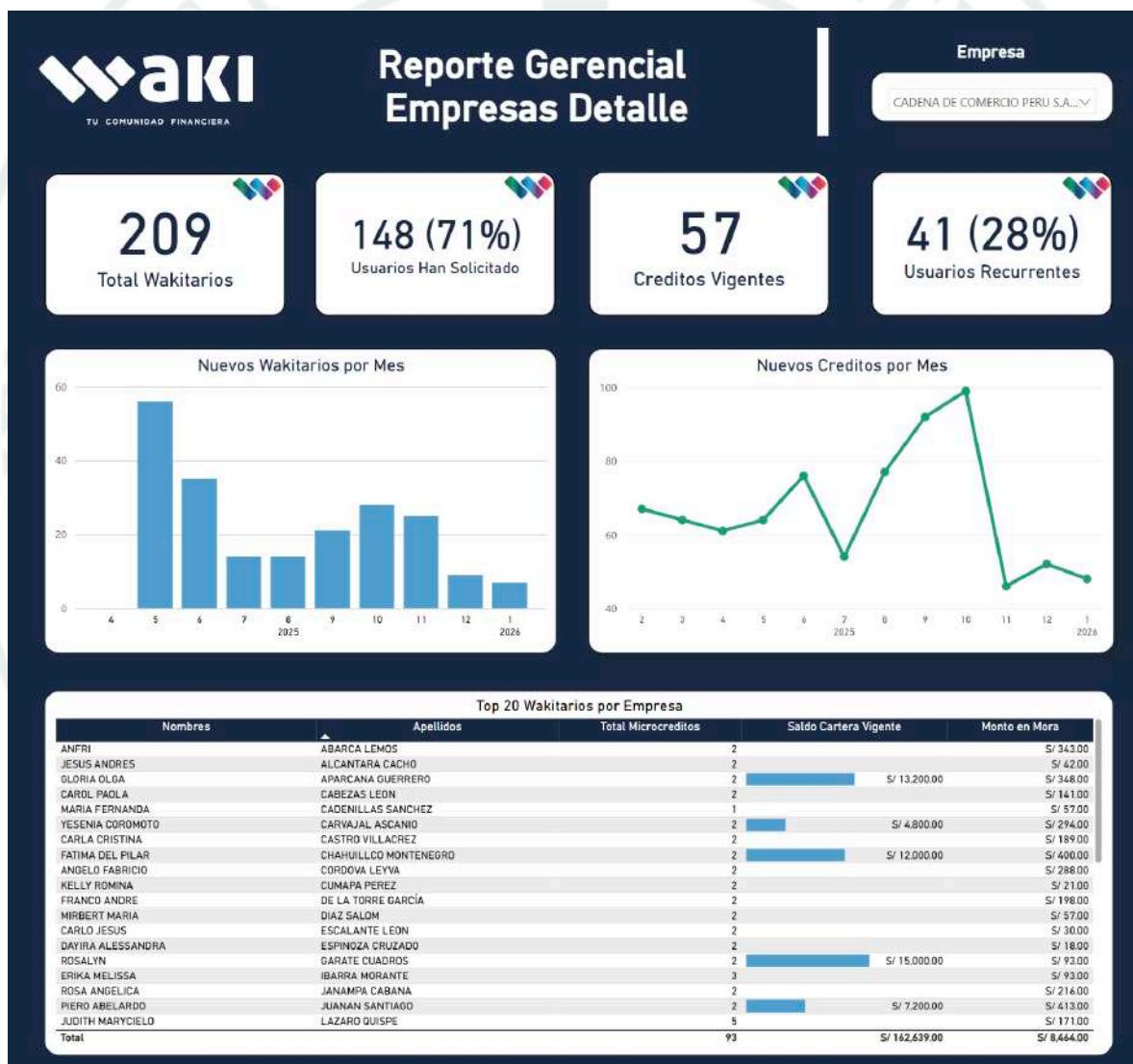


Nota. Elaboración propia.

Esta página presenta una vista consolidada de todas las empresas con créditos activos. Incluye cuatro indicadores clave (KPIs) implementados como tarjetas: Saldo Cartera Vigente, Total Wakitarios, Créditos Vigentes y Empresas Activas. Complementan la vista un gráfico de barras horizontales con el Top 10 de empresas por saldo de cartera y una tabla de ranking que muestra por cada empresa: total de wakitarios, créditos vigentes, tasa de morosidad y saldo de cartera vigente.

2.6.2.2. Página 2: Reporte Gerencial - Empresas Detalle

Figura 18: Dashboard Gerencial - Detalle por Empresa



Nota. Elaboración propia.

Permite analizar el detalle de una empresa mediante un filtro desplegable. Presenta cuatro KPIs: Total Wakitarios, Usuarios que Han Solicitado crédito, Créditos Vigentes y Usuarios Recurrentes. Incluye gráficos de evolución mensual de nuevos wakitarios y nuevos créditos, y una tabla con el Top 20 de wakitarios por cantidad de microcréditos solicitados.

2.6.2.3. Página 3: Reporte Operativo - Cobranza

Figura 19: Dashboard Operativo - Cobranza

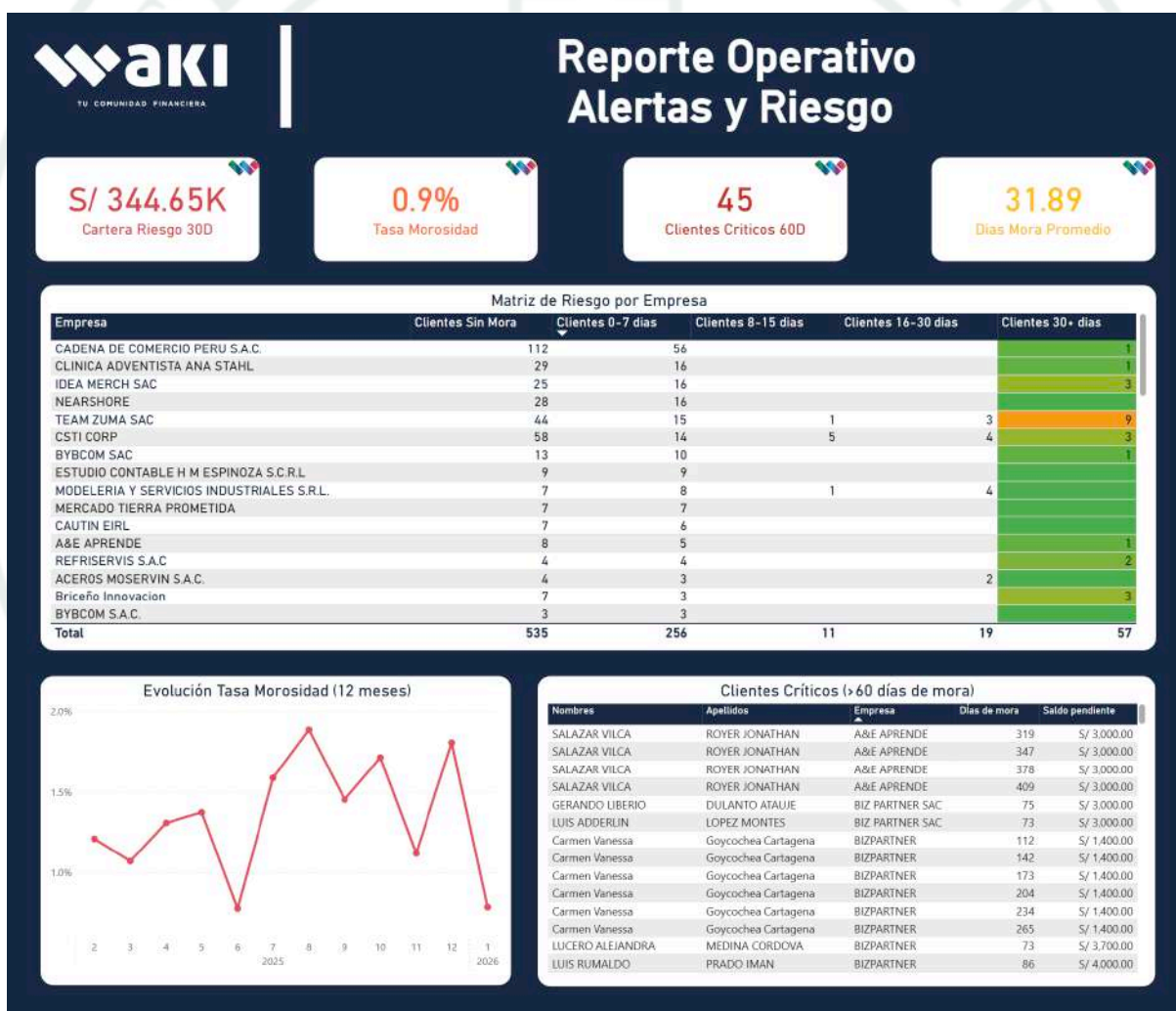


Nota. Elaboración propia.

Orientada al seguimiento diario de cobranza. Presenta cuatro KPIs críticos: Cuotas Vencidas (cantidad de cuotas pendientes de pago), Cuotas Por Vencer en 7 días, Clientes con más de 15 días de mora, y Monto Total de Cuotas Vencidas. Incluye un gráfico de columnas que muestra la distribución de clientes por rangos de días de mora (0-7, 8-15, 16-30, 31-60, +60 días) con código de colores semafórico (verde a rojo según severidad). La tabla principal lista todas las cuotas vencidas con detalle de cliente, empresa, número de cuota, monto, días de mora y fecha de vencimiento.

2.6.2.4. Página 4: Reporte Operativo - Alertas y Riesgo

Figura 20: Dashboard Operativo - Alertas y Riesgo



Nota. Elaboración propia.

Enfocada en riesgo crediticio. Presenta cuatro KPIs: Cartera en Riesgo, Tasa de Morosidad, Clientes Críticos (+60 días de mora) y Promedio de Días de Mora. Incluye una matriz de riesgo por empresa, un gráfico con la evolución de morosidad en los últimos 12 meses y una tabla de clientes críticos ordenados por días de atraso.

2.6.3. Validación con el usuario final

Los entregables funcionales del proyecto fueron revisados y aprobados por la administradora de Holding Waki SAC, quien es la usuaria principal de la solución y responsable directa de los reportes operativos y gerenciales. La conformidad cubre dos aspectos fundamentales del diseño:

- **Indicadores clave de desempeño (KPIs):** la administradora declaró que los KPIs implementados en los cuatro dashboards de Power BI corresponden efectivamente a la información que requiere para el seguimiento operativo y gerencial de la cartera de microcréditos. El acta de aprobación se incluye en el Anexo B.
- **Diseño visual de los dashboards:** la administradora declaró que el diseño visual, la distribución de los elementos, los gráficos y el formato general de los cuatro dashboards son adecuados para su uso diario. El acta de aprobación se incluye en el Anexo C.

Estas conformidades sustentan la pertinencia funcional de la solución desde la perspectiva del usuario final.

2.7. Resumen del capítulo

Este capítulo documentó el desarrollo completo de la solución de inteligencia de negocios para el seguimiento de microcréditos en Holding Waki SAC, desde la justificación metodológica hasta la validación funcional con el usuario.

Se adoptó la metodología de Ralph Kimball por su enfoque bottom-up orientado al negocio, que permitió construir un datamart de microcréditos con entregas incrementales y resultados tangibles en tiempos reducidos.

El análisis funcional identificó dos perfiles de usuarios diferenciados: el equipo operativo (administradora y analista de créditos) que requiere información detallada y actualizada sobre el estado diario de los microcréditos, y la gerencia (junta de socios) que necesita indicadores consolidados para decisiones estratégicas. Esta separación guió el diseño de los cuatro dashboards implementados.

El modelo dimensional se estructuró como un esquema estrella con granularidad a nivel de cuota de microcrédito, permitiendo responder tanto a consultas operativas (cuotas vencidas, próximos vencimientos) como gerenciales (evolución de cartera, tasa de morosidad). Las cinco dimensiones definidas (cliente, crédito, tiempo, producto, estado de pago) cubren las perspectivas de análisis requeridas por el negocio.

La arquitectura implementada en AWS integra funciones Lambda serverless para el proceso ETL, Amazon Redshift Serverless como data warehouse y Power BI para la visualización, con orquestación automática mediante EventBridge. La conexión entre Power BI y Redshift se resolvió mediante una solución híbrida que combina túneles SSM para el entorno de desarrollo y un On-premises Data Gateway para producción, manteniendo la seguridad del data warehouse en una VPC privada sin exposición pública.





CAPÍTULO III

RESULTADOS

3.1. Introducción

El presente capítulo evalúa el impacto de la solución implementada mediante la comparación de las cuatro métricas definidas en la línea base del Capítulo 1: tiempo de elaboración de reportes, oportunidad de la información, exactitud de la información y retrabajo sobre el proceso. Para cada métrica se presenta el valor registrado antes de la implementación, el valor observado después y la evidencia objetiva que sustenta el resultado.

La evaluación sigue un diseño pre-experimental de un único grupo con medición antes y después de la implementación, dado que Holding Waki SAC cuenta con un solo equipo de usuarios y no existe un contexto organizacional comparable que permita conformar un grupo de control. Este diseño resulta adecuado para el alcance del proyecto: las mejoras cuantificadas se atribuyen directamente a la automatización implementada, dado que no se produjeron otros cambios tecnológicos ni organizacionales durante el período de evaluación.

3.2. Tiempo de elaboración de reportes

La primera métrica evaluada corresponde al tiempo total que insume la elaboración de los dos reportes periódicos definidos en la línea base: el reporte operativo (semanal) y el reporte gerencial (cada 10 días). La Tabla 20 presenta la comparación entre el proceso manual previo y el proceso automatizado actual.

En la línea base, el tiempo de elaboración se medía como el tiempo que la administradora dedicaba a construir manualmente cada reporte en Excel. Con la solución implementada, ese trabajo desaparece: los datos ya están cargados en el Data Warehouse gracias al proceso ETL automatizado (cuyo rendimiento interno se documenta en el Capítulo 2), y los reportes se consumen directamente como dashboards publicados en Power BI Service. En consecuencia, el tiempo de elaboración del reporte pasa a medirse como el **tiempo promedio que tarda el usuario final en acceder al dashboard desde que lo abre hasta que los indicadores están visibles y listos para consulta**.

Para cuantificar este nuevo escenario, se cronometró en cinco ocasiones el tiempo de carga del dashboard publicado en Power BI Service, obteniendo un promedio aproximado de **7 segundos**. Dado que los reportes operativo y gerencial ahora conviven en el mismo archivo .pbix organizado en cuatro páginas, ambos comparten el mismo tiempo de acceso. La Tabla 20 presenta la comparación.

Tabla 20: Comparación del tiempo de elaboración de reportes antes y después de la implementación

Reporte	Antes	Después	Mejora
Operativo	480 min (1 día)	~7 s (carga del dashboard)	99.98 %
Gerencial	720 min (1.5 días)	~7 s (carga del dashboard)	99.98 %

Nota. Elaboración propia.

El tiempo de elaboración del reporte, medido desde la experiencia del usuario final, pasó de entre 1 y 1.5 días hábiles de trabajo manual a un promedio de 7 segundos de carga del dashboard publicado. La mejora es del 99.98 % para ambos reportes y refleja que la solución reemplaza por completo la tarea de elaboración: la administradora deja de dedicar tiempo a construir los reportes y los usuarios simplemente consumen la información ya disponible en Power BI Service desde cualquier dispositivo con conexión a internet.

3.3. Oportunidad de la información

La segunda métrica evaluada corresponde a la oportunidad con la que la información llega a sus destinatarios: la administradora, el analista de créditos y la junta de socios. La Tabla 21 presenta la comparación entre el estado previo y el actual.

Tabla 21: Comparación de la oportunidad de la información antes y después de la implementación

Aspecto	Antes	Después	Mejora
Frecuencia de actualización	Semanal / Manual	Diaria / Automática	Diaria
Retraso promedio de entrega	2.3–3.5 días	0 días	100 %
Ventana de decisión perdida	32.9%–35.0 % del ciclo	0 % del ciclo	100 %

Nota. Elaboración propia.

3.3.1. Evidencia objetiva de la actualización automática

La oportunidad en la entrega de información se sustenta en dos evidencias complementarias: la configuración del evento en Amazon EventBridge Scheduler, y el historial real de ejecuciones diarias registrado en Amazon CloudWatch Logs.

Las seis funciones Lambda están configuradas para ejecutarse diariamente en horarios escalonados de la madrugada (zona horaria America/Lima), garantizando que los dashboards de Power BI cuenten con datos actualizados antes del inicio del horario laboral. La Figura 6, presentada en el Capítulo 2, muestra la configuración de los *schedules* en la consola de AWS EventBridge Scheduler, evidenciando que la actualización de datos no depende de intervención manual.

Adicionalmente, la Figura 21 presenta el listado de *log streams* generados por la función *etl-fact-cuota* en Amazon CloudWatch Logs. Cada *log stream* corresponde a una ejecución diaria del proceso ETL completo, ya que esta función se encuentra al final del pipeline (las dimensiones se procesan antes). El registro consecutivo de ejecuciones demuestra que el ETL efectivamente corre todos los días tal como está programado en EventBridge.

Figura 21: Historial de ejecuciones diarias de etl-fact-cuota en CloudWatch Logs

The screenshot shows the 'Log streams (89)' view in AWS CloudWatch. It includes a search bar, filter options for 'Exact match' and 'Show expired', and a table of log streams. The table has two columns: 'Log stream' and 'Last event time'. The log streams listed are:

Log stream	Last event time
2026/04/10/[\$LATEST]18950bd746f254f0fadbe1b9aa5978393	2026-04-10 02:00:25 (UTC-05:00)
2026/04/09/[\$LATEST]2a98f57d8abc4f7b88be107611a6bdee	2026-04-09 02:00:25 (UTC-05:00)
2026/04/08/[\$LATEST]83088497cb614855b930499d240bb04	2026-04-08 02:00:25 (UTC-05:00)
2026/04/07/[\$LATEST]d720e6a84108402abe2455c8f135d1f4	2026-04-07 02:00:25 (UTC-05:00)
2026/04/06/[\$LATEST]d49e76f47d9c4f028ac0eb95070aaa6a	2026-04-06 02:00:25 (UTC-05:00)
2026/04/05/[\$LATEST]5a969f4baaac4c24a5f163e3b477879a	2026-04-05 02:00:26 (UTC-05:00)
2026/04/04/[\$LATEST]385c232e299b43619cb47724bb57e915	2026-04-04 02:00:24 (UTC-05:00)
2026/04/03/[\$LATEST]9a5331d8e9f74434956b6a2bf2e9ec72	2026-04-03 02:00:25 (UTC-05:00)
2026/04/02/[\$LATEST]9b03f5b7427d412c9533dc3538456aff	2026-04-02 02:00:25 (UTC-05:00)
2026/04/01/[\$LATEST]e021b36afa794c779b434bc1dc6e6a3a	2026-04-01 02:00:25 (UTC-05:00)
2026/03/31/[\$LATEST]cb2d335239bd446eb31e52f1ccff3af6	2026-03-31 02:00:24 (UTC-05:00)
2026/03/30/[\$LATEST]44c949187fd84dd9ba82940c557b07e1	2026-03-30 02:00:25 (UTC-05:00)
2026/03/29/[\$LATEST]15787d20cb784a8ea6db41b0cd9f958	2026-03-29 02:00:26 (UTC-05:00)

Nota. Elaboración propia.

La solución transformó el modelo de entrega de información de manual a automático. Los dashboards publicados en Power BI Service están disponibles desde cualquier dispositivo con conexión a internet durante el horario laboral, eliminando completamente los retrasos de 2.3 a 3.5 días identificados en la línea base. En términos de la ventana de decisión perdida, el valor pasa del 32.9%–35.0% del ciclo registrado antes de la implementación a 0% del ciclo, ya que la información pasa a estar disponible al inicio de cada jornada laboral de forma consistente y deja de depender del tiempo que la administradora pueda dedicar a la elaboración manual.

3.4. Exactitud de la información

La tercera métrica evaluada corresponde a la exactitud de la información contenida en los reportes. La Tabla 22 presenta la comparación entre la tasa de errores observada en la línea base (a partir de un muestreo de 12 reportes operativos y 9 reportes gerenciales de los últimos tres meses) y la tasa observada después de la implementación.

Tabla 22: Comparación de la exactitud de la información antes y después de la implementación

Tipo de error	Antes	Después	Mejora
Reportes con errores	25.0%–33.3 %	0 %	100 %
Errores de cálculo	8.3%–22.2 %	0 %	100 %
Errores de transcripción	16.7%–11.1 %	0 %	100 %

Nota. Elaboración propia.

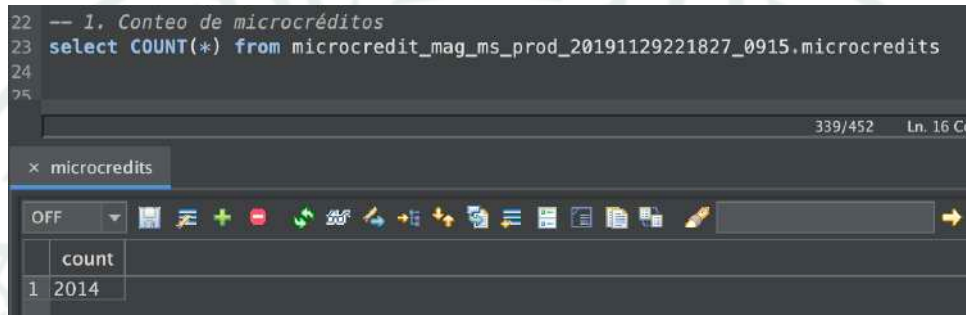
El Data Warehouse almacena únicamente datos provenientes de Cassandra (registros individuales de microcréditos, cuotas y clientes), sin realizar cálculos propios (campos calculados): los indicadores agregados que consumen los usuarios (saldos de cartera, tasa de morosidad, conteos de clientes, etc.) se computan dentro de Power BI mediante medidas DAX definidas sobre el modelo dimensional. Por esta razón, la exactitud de los valores que el usuario final observa en los dashboards depende de dos condiciones: que los datos granulares cargados en el Data Warehouse reflejen fielmente a la fuente, y que las medidas DAX implementadas sobre esos datos sean correctas. La exactitud del producto final se sustenta, entonces, en dos evidencias complementarias:

1. **Reconciliación de volúmenes entre Cassandra y Redshift.** Verifica que la totalidad de los microcréditos, cuotas y clientes existentes en la fuente se materializan íntegramente en el Data Warehouse sin pérdida ni duplicación. Se presenta a continuación.
2. **Conformidad funcional de los KPIs con la administradora.** Los valores que los usuarios ven en los dashboards son producto de las medidas DAX ejecutadas sobre el modelo dimensional en Power BI. La administradora del área aprobó formalmente los KPIs implementados en los cuatro dashboards (Anexo B), confirmando que los valores mostrados corresponden a la información que ella requiere y reconoce como correcta para el seguimiento de la cartera.

3.4.1. Reconciliación de volúmenes entre Cassandra y Redshift

Para verificar que el proceso ETL transfiera íntegramente la información desde la base de datos fuente hacia el Data Warehouse, se ejecutaron consultas de conteo (COUNT(*)) en ambos extremos del flujo de datos para las tres entidades principales del modelo dimensional: microcréditos, cuotas y clientes. Las Figuras 22 a 27 presentan las capturas de las consultas ejecutadas en cada motor de base de datos.

Figura 22: Conteo de microcréditos en Cassandra

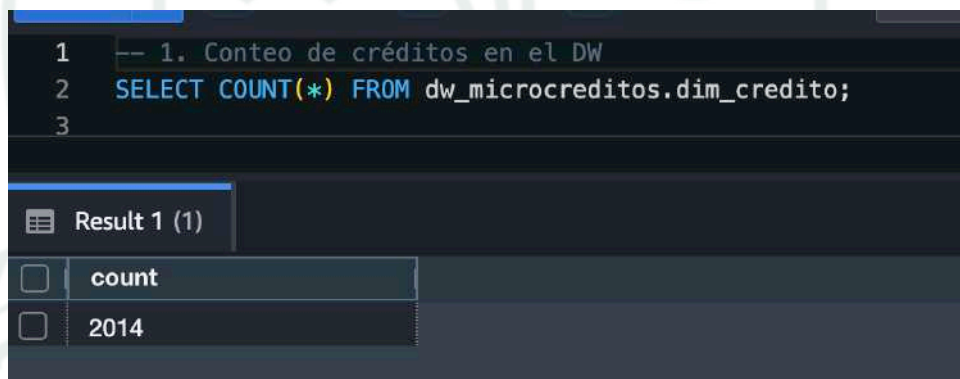


```
22 -- 1. Conteo de microcréditos
23 select COUNT(*) from microcredit_mag_ms_prod_20191129221827_0915.microcredits
24
25
```

count
1 2014

Nota. Elaboración propia.

Figura 23: Conteo de microcréditos en Redshift (dw_microcreditos.dim_credito)

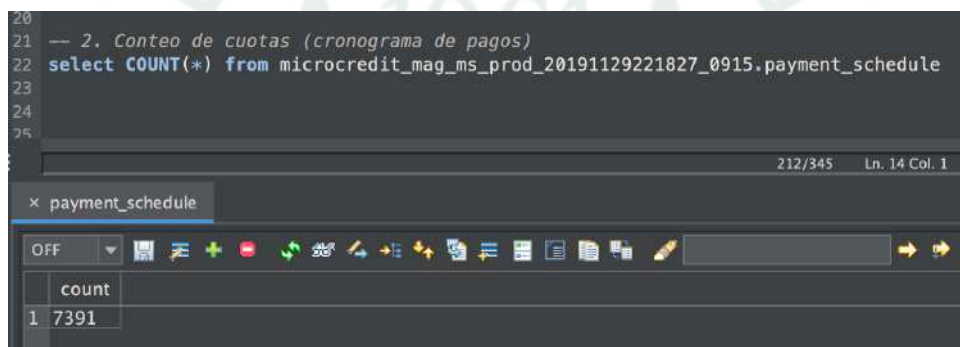


```
1 -- 1. Conteo de créditos en el DW
2 SELECT COUNT(*) FROM dw_microcreditos.dim_credito;
3
```

count
2014

Nota. Elaboración propia.

Figura 24: Conteo de cuotas en Cassandra (payment_schedule)



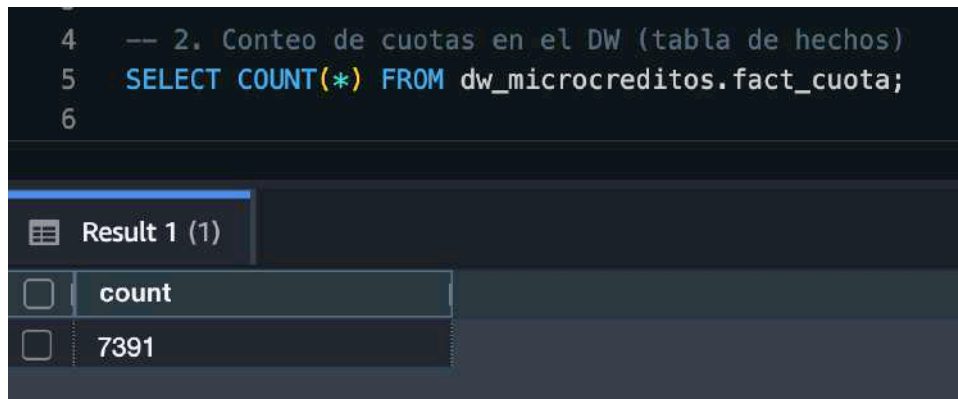
```
20
21 -- 2. Conteo de cuotas (cronograma de pagos)
22 select COUNT(*) from microcredit_mag_ms_prod_20191129221827_0915.payment_schedule
23
24
25
```

count
1 7391

Nota. Elaboración propia.

Figura 25: Conteo de cuotas en Redshift (dw_microcreditos.fact_cuota)

```
4 -- 2. Conteo de cuotas en el DW (tabla de hechos)
5 SELECT COUNT(*) FROM dw_microcreditos.fact_cuota;
6
```



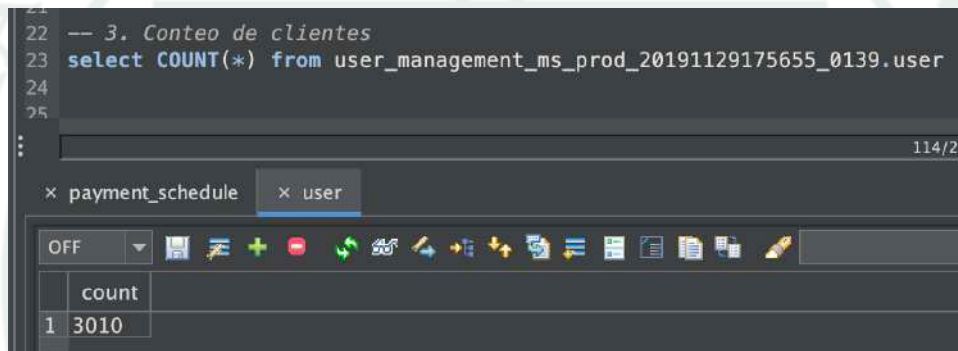
The screenshot shows a SQL query in Redshift that counts the number of records in the fact_cuota table. The result is displayed in a table with one row and one column.

count
7391

Nota. Elaboración propia.

Figura 26: Conteo de usuarios en Cassandra (user_management_ms.user)

```
22 -- 3. Conteo de clientes
23 select COUNT(*) from user_management_ms_prod_20191129175655_0139.user
24
25
```



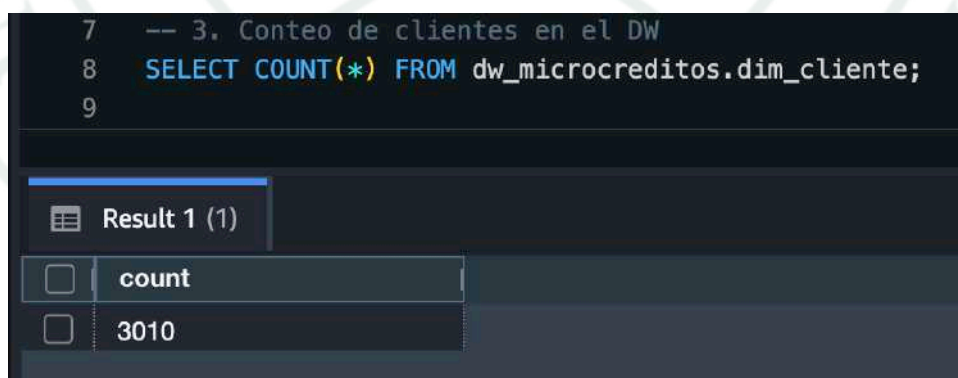
The screenshot shows a SQL query in Cassandra that counts the number of records in the user table. The result is displayed in a table with one row and one column.

count
3010

Nota. Elaboración propia.

Figura 27: Conteo de clientes en Redshift (dw_microcreditos.dim_cliente)

```
7 -- 3. Conteo de clientes en el DW
8 SELECT COUNT(*) FROM dw_microcreditos.dim_cliente;
9
```



The screenshot shows a SQL query in Redshift that counts the number of records in the dim_cliente table. The result is displayed in a table with one row and one column.

count
3010

Nota. Elaboración propia.

La Tabla 23 consolida los resultados obtenidos en ambos motores de base de datos.

Tabla 23: Reconciliación de volúmenes entre Cassandra y Redshift

Entidad	Cassandra	Redshift	Diferencia
Microcréditos	2 014	2 014	0
Cuotas	7 391	7 391	0
Clientes	3 010	3 010	0

Nota. Elaboración propia.

Los conteos obtenidos en ambos extremos del flujo coinciden de forma exacta para las tres entidades evaluadas, confirmando la integridad de la información cargada en el Data Warehouse, sin pérdida ni duplicación de registros.

La eliminación de los procesos manuales de transcripción y cálculo erradicó los errores identificados en la línea base (entre 25.0% y 33.3% de reportes con errores). La combinación de pruebas unitarias que validan la lógica de transformación del ETL, la reconciliación de volúmenes que verifica la integridad de la carga en el Data Warehouse y la aprobación funcional de los KPIs por parte de la administradora respalda la ausencia de errores en los valores que los usuarios consumen a través de los dashboards, con la salvedad de que el resultado depende de que se mantengan las condiciones operativas y el esquema de datos de origen.

3.5. Retrabajo sobre el proceso

La cuarta métrica evaluada corresponde al retrabajo asociado a la elaboración de los reportes. Como se documentó en la línea base (Capítulo 1, Tabla 5), la administradora solicitaba al asistente del área una revisión adicional de cada reporte antes de distribuirlo, con el fin de detectar y corregir los errores que introducía el proceso manual. Este retrabajo consumía 51 horas al mes del asistente y representaba un costo económico cuantificable a partir de su remuneración. La Tabla 24 presenta la comparación entre el estado previo y el actual.

Tabla 24: Comparación del retrabajo sobre el proceso antes y después de la implementación

Aspecto	Antes	Después	Mejora
Horas mensuales de retrabajo	51 h	0 h	100 %
Costo mensual del retrabajo	S/. 478.13	S/. 0	100 %
Costo anual del retrabajo	S/. 5 737.50	S/. 0	100 %

Nota. Elaboración propia.

La eliminación del proceso manual erradicó el retrabajo asociado a la revisión de reportes, liberando 51 horas mensuales del asistente del área y representando un ahorro equivalente a S/. 478.13 mensuales (S/. 5 737.50 al año), calculado a partir del sueldo de referencia y la jornada laboral estándar. Esta métrica refuerza el impacto económico de la solución además de permitir al asistente dedicar las horas a actividades propias de su puesto con mayor valor para el área.

3.6. Conclusiones del capítulo

En términos cuantitativos, la publicación de los reportes como dashboards en Power BI Service redujo el tiempo de elaboración a un promedio de siete segundos de carga, lo que representa mejoras del 99.98 % respecto de los 480 minutos del reporte operativo y los 720 minutos del reporte gerencial registrados en la línea base. La oportunidad de la información pasó de un retraso promedio de 2.3 a 3.5 días hábiles (ventana de decisión perdida del 32.9%–35.0 % del ciclo) a entrega diaria automática (0% de ventana perdida). La exactitud de los valores consumidos en los dashboards alcanzó el 0 % de errores, respaldada por la integridad verificada del ETL, la reconciliación de volúmenes entre Cassandra y Redshift, y la aprobación formal de los KPIs por parte de la administradora. Finalmente, el retrabajo asociado a la revisión manual se redujo de 51 horas mensuales a 0, liberando un equivalente aproximado de S/. 5 737.50 anuales en horas del asistente del área.

En términos cualitativos, la solución reemplazó un proceso manual por un flujo automatizado, disponible desde cualquier dispositivo con conexión a internet durante el horario laboral, además de liberar horas remuneradas del equipo que pueden redirigirse a actividades de mayor valor.



CONCLUSIONES Y RECOMENDACIONES

3.7. Conclusiones

A partir del desarrollo e implementación de la solución de inteligencia de negocios para el seguimiento de microcréditos en Holding Waki SAC, se presentan las siguientes conclusiones en función de los objetivos específicos planteados:

1. **Respecto al diseño e implementación del proceso ETL:** Se diseñó e implementó un proceso ETL automatizado con seis funciones AWS Lambda que extrae datos desde Apache Cassandra, aplica transformaciones con Pandas y carga el datamart en Amazon Redshift Serverless, completando el procesamiento diario de las dimensiones y la tabla de hechos en aproximadamente once minutos sin intervención humana.
2. **Respecto al desarrollo del Data Warehouse:** Se desarrolló un Data Warehouse en Amazon Redshift Serverless bajo la metodología Kimball, implementando un esquema estrella con granularidad a nivel de cuota, cinco dimensiones y una tabla de hechos que consolida datos operativos y gerenciales de los microcréditos de Holding Waki SAC.
3. **Respecto a la creación de dashboards:** Se crearon cuatro dashboards en Power BI con 31 medidas DAX que visualizan métricas clave de la cartera, diferenciando vistas gerenciales y operativas, con actualización diaria automática y acceso desde cualquier dispositivo con conexión a internet durante el horario laboral.
4. **Respecto a la evaluación del impacto:** La solución redujo el tiempo de elaboración de reportes en 99.98 % (de 480 y 720 minutos a aproximadamente 7 segundos de carga del dashboard), eliminó los retrasos de entrega de 2.3 a 3.5 días, los errores de datos (de 25.0–33.3 % a 0 %) y el retrabajo de revisión manual (51 horas mensuales del asistente del área, equivalentes a aproximadamente S/. 5 737.50 anuales).

En conjunto, los resultados obtenidos confirman que la implementación de una solución BI serverless en AWS optimizó la generación de reportes para el seguimiento de microcréditos de Holding Waki SAC, demostrando la viabilidad y efectividad de esta arquitectura para organizaciones de microfinanzas de tamaño reducido.

3.8. Recomendaciones

Con base en la experiencia obtenida durante el desarrollo e implementación del proyecto, se presentan las siguientes recomendaciones para trabajo futuro y mejora continua de la solución:

3.8.1. Recomendación 1: Expansión del modelo dimensional

Se recomienda desarrollar datamarts adicionales para procesos no cubiertos actualmente: análisis de solicitudes rechazadas, seguimiento de desembolsos y análisis de reprogramaciones. La metodología Kimball facilita esta expansión mediante las dimensiones conformadas existentes (dim_cliente, dim_tiempo, dim_producto).

3.8.2. Recomendación 2: Implementación de alertas automáticas

Se recomienda implementar alertas automáticas mediante funciones Lambda adicionales que consulten Redshift tras cada carga ETL y utilicen Amazon SNS para enviar notificaciones ante situaciones críticas, como incrementos de morosidad, clientes con más de 60 días de atraso o cuotas próximas a vencer.

3.8.3. Recomendación 3: Desarrollo de modelos predictivos

Se recomienda aprovechar el historial estructurado del Data Warehouse para entrenar modelos de machine learning en Amazon SageMaker, orientados a la predicción de morosidad, segmentación de clientes por comportamiento de pago y detección de clientes con alta probabilidad de abandono.

REFERENCIAS

- Administración de Holding Waki S.A.C. (2025). *Reuniones de levantamiento de información con la administradora de holding waki s.a.c.* Comunicación personal. (Reuniones sostenidas con la administradora durante el segundo semestre de 2025 para el levantamiento de procesos, tiempos y métricas de elaboración de reportes)
- Amazon Web Services. (2025a). *Amazon eventbridge*. Descargado de <https://aws.amazon.com/es/eventbridge/> (Accedido el 20 de diciembre de 2025)
- Amazon Web Services. (2025b). *Amazon redshift*. Descargado de <https://aws.amazon.com/es/redshift/> (Accedido el 20 de diciembre de 2025)
- Amazon Web Services. (2025c). *Aws lambda*. Descargado de <https://aws.amazon.com/es/lambda/> (Accedido el 20 de diciembre de 2025)
- Anugerah, C. S. T. (2022). Data warehouse design for sales transactions on. *Jurnal CoreIT*, 8(2).
- Apolaya Segura, C. E., y Flores Suárez, M. D. (2022). Digitalización de sala situacional utilizando power bi y azure sql database para mejorar el monitoreo de covid-19 en etapa de emergencia-red asistencial essalud lambayeque.
- Berisha, B., Mëziu, E., y Shabani, I. (2022). Big data analytics in cloud computing: an overview. *Journal of Cloud Computing*, 11(1), 24.
- Chávez Rengifo, K., y Amaya Pacheco, O. J. (2023). Solución de big data para el área de comercialización de la empresa inversiones santa maría en el período 2021 bajo el ecosistema de apache hadoop y microsoft azure.
- Damiano Espinoza, K. P. (2020). Implementación de un data mart para mejorar la toma de decisiones en el área de continuidad del negocio, banco ripley.
- Daza, A., De la Paz Tito, E. M., y Cajas, Y. R. A. (2024). Impact of business intelligence on incident management in the control center of a security company. *Bulletin of Electrical Engineering and Informatics*, 13(1), 422–435.
- Frankenfield, J. (2024). *Business intelligence (bi)*. Descargado de <https://www.investopedia.com/terms/b/business-intelligence-bi.asp> (Accedido el 20 de diciembre de 2025)
- Group, K. (2008). *Fact tables*. Descargado de <https://www.kimballgroup.com/2008/11/fact-tables/> (Accedido el 20 de diciembre de 2025)
- Group, K. (2023). *Life cycle methodology*. Descargado de <https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/dw-bi-lifecycle-method/> (Accedido el 20 de diciembre de 2025)
- Group, K. (2024). *Dimension tables*. Descargado de <https://www.kimballgroup.com/fundamentals/dimension-table-structure/> (Accedido el 20 de diciembre de 2025)
- Hinojosa Zarate, L. R. (2023). Business intelligence para la toma de decisiones en el área banco de datos en una institución gubernamental, lima 2023.

- Holding Waki S.A.C. (2024). *Nosotros — waki*. Descargado de <https://www.waki.com.pe/#Nosotros> (Accedido el 10 de enero de 2026)
- Holding Waki S.A.C. (2025). *Historial de entregas de reportes operativos y gerenciales*. Correos electrónicos internos de la empresa. (Registros de los últimos tres meses utilizados como base para el análisis de oportunidad y exactitud de los reportes)
- IBM. (2024). *What is etl?* Descargado de <https://www.ibm.com/cloud/learn/etl> (Accedido el 20 de diciembre de 2025)
- Inmon, W. H., Strauss, D., y Neushloss, G. (2010). *Dw 2.0: The architecture for the next generation of data warehousing*. Elsevier.
- Kanchepu, N. (2023). Digital transformation in banking industry: cloud computing as a key enabler. *International Numeric Journal of Machine Learning and Robots*, 7(7).
- Kimball, R., Ross, M., Thornthwaite, W., Mundy, J., y Becker, B. (2008). *The data warehouse lifecycle toolkit*. John Wiley & Sons.
- Luengo Herrero, A., y cols. (2023). Desarrollo de una aplicación basada en procedimientos etl/elt para la generación de indicadores sobre procesos productivos en el ámbito de las smart factories.
- Microsoft. (2025). *Power bi overview*. Descargado de <https://docs.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview> (Accedido el 20 de diciembre de 2025)
- Mora, G. (2020). Influencia de la inteligencia de negocios en los procesos de toma de decisiones dentro de las instituciones financieras. *Realidad Empresarial*(10), 21–24.
- Moyano, D. M. B., y Molina, M. A. C. (2020). Implementación de data mart, en power bi, para el análisis de ventas a clientes, en los eonegocios “gransol”. *Polo del Conocimiento: Revista científico-profesional*, 5(1), 647–673.
- Pamo Ferro, D. A. (2022). Solución de inteligencia de negocios utilizando la metodología ralph kimball en el proceso de toma de decisiones en el área de almacén de la empresa gnecco y cia sac-2022.
- Peralta Villasante, G. Y. (2021). Data mart para el proceso de toma de decisiones en área de ventas para la empresa de transportes reyna.
- Pratt, M. K. (2023). *Data warehouse*. Descargado de <https://searchdatamanagement.techtarget.com/definition/data-warehouse> (Accedido el 20 de diciembre de 2025)
- Ramos Ramos, G. Y. (2023). Optimización del proceso de seguimiento y control de materiales core mediante la aplicación de inteligencia de negocios en una empresa minera en el sur del Perú.
- Retuerto, M., Tuero, B., y Andrade-Arenas, L. (2023, 04). Business intelligence implementation using power bi for decision making in peruvian banking systems. *International Journal of Engineering Trends and Technology*, 71, 97-108. doi: 10.14445/22315381/IJETT-V71I4P209
- Vinoth, S., Vemula, H. L., Haralayya, B., Mamgain, P., Hasan, M. F., y Naved, M. (2022). Application of cloud computing in banking and e-commerce and related security threats.



ANEXOS

Anexo A: Documentación técnica de la solución

A.1. Introducción

El presente anexo reúne la documentación técnica completa de la solución implementada para el seguimiento de microcréditos en Holding Waki SAC. Se proporciona información detallada sobre la estructura del código fuente, la configuración del *data warehouse*, diccionarios de datos completos y los procedimientos de despliegue del sistema.

A.2. Estructura del proyecto

A.2.1. Organización del repositorio

El código fuente del ETL se organiza en un repositorio Git con la siguiente estructura de directorios:

```
1 waki-etl/
2 |-- Dockerfile           # Imagen Docker para Lambda
3 |-- README.md           # Documentación del proyecto
4 |-- layer/
5 | |-- requirements.txt   # Dependencias Python
6 |-- shared/             # Módulos compartidos
7 | |-- __init__.py
8 | |-- aws_utils.py      # Utilidades AWS/Secrets Manager
9 | |-- cassandra_client.py # Cliente Cassandra
10 | |-- redshift_client.py # Cliente Redshift
11 |-- etl-dim-tiempo/
12 | |-- handler.py        # Handler dimensión tiempo
13 |-- etl-dim-estado-pago/
14 | |-- handler.py        # Handler dimensión estado pago
15 |-- etl-dim-producto/
16 | |-- handler.py        # Handler dimensión producto
17 |-- etl-dim-cliente/
18 | |-- handler.py        # Handler dimensión cliente
19 |-- etl-dim-credito/
20 | |-- handler.py        # Handler dimensión crédito
```

```
21 |-- etl-fact-cuota/  
22 | |-- handler.py          # Handler tabla de hechos
```

Listing A.1: Estructura de directorios del proyecto waki-etl

A.2.2. Dependencias del proyecto

Las dependencias Python se gestionan mediante un archivo requirements.txt que se instala durante la construcción de la imagen Docker:

```
1 # Cassandra driver  
2 cassandra-driver==3.29.1  
3  
4 # Redshift connector  
5 redshift-connector==2.1.0  
6  
7 # Data processing  
8 pandas==2.1.4  
9 numpy==1.26.2  
10  
11 # AWS SDK  
12 boto3==1.34.0
```

Listing A.2: Dependencias Python (requirements.txt)

A.2.3. Dockerfile

La imagen Docker se construye a partir de la imagen base oficial de AWS Lambda para Python 3.11:

```
1 FROM public.ecr.aws/lambda/python:3.11  
2  
3 COPY layer/requirements.txt ${LAMBDA_TASK_ROOT}  
4  
5 RUN pip install -r requirements.txt  
6  
7 COPY shared ${LAMBDA_TASK_ROOT}/shared  
8  
9 COPY etl-dim-cliente ${LAMBDA_TASK_ROOT}/etl-dim-cliente  
10 COPY etl-dim-credito ${LAMBDA_TASK_ROOT}/etl-dim-credito  
11 COPY etl-dim-estado-pago ${LAMBDA_TASK_ROOT}/etl-dim-estado-pago
```

```

12 COPY etl-dim-producto ${LAMBDA_TASK_ROOT}/etl-dim-producto
13 COPY etl-dim-tiempo ${LAMBDA_TASK_ROOT}/etl-dim-tiempo
14 COPY etl-fact-cuota ${LAMBDA_TASK_ROOT}/etl-fact-cuota
15
16 CMD [ "etl-dim-tiempo.handler.handler" ]

```

Listing A.3: Dockerfile para despliegue en AWS Lambda

El comando CMD por defecto apunta a etl-dim-tiempo, pero al crear cada función Lambda se configura el handler específico mediante la variable de entorno AWS_LAMBDA_FUNCTION_HANDLER.

A.3. Módulos compartidos

A.3.1. Cliente Cassandra

El módulo `cassandra_client.py` implementa la conexión y extracción de datos desde Apache Cassandra:

```

1 class CassandraClient:
2     """Cliente de Cassandra usado para configurar."""
3
4     def __init__(self):
5         self.cluster = None
6         self.session = None
7         self._connect()
8
9     def _connect(self):
10        """Establece conexión con el clúster de Cassandra"""
11        contact_points = os.environ.get('CASSANDRA_CONTACT_POINTS', '').split(',')
12        datacenter = os.environ.get('CASSANDRA_DATACENTER', 'AWS_VPC_WAKI_PROD_
↳ EAST1')
13        username = os.environ.get('CASSANDRA_USERNAME', '')
14        password = os.environ.get('CASSANDRA_PASSWORD', '')
15
16        auth_provider = PlainTextAuthProvider(
17            username=username,
18            password=password
19        )
20
21        self.cluster = Cluster(
22            contact_points=contact_points,

```

```

23     port=9042,
24     auth_provider=auth_provider,
25     protocol_version=4,
26     load_balancing_policy=DCAwareRoundRobinPolicy(local_dc=datacenter)
27 )
28
29     self.session = self.cluster.connect()
30
31     def fetch_all(self, keyspace: str, table: str,
32                 columns: List[str] = None,
33                 where_clause: str = None) -> List[Dict[str, Any]]:
34         """Obtiene todas las filas de una tabla"""
35         cols_str = ", ".join(columns) if columns else "*"
36         query = f"SELECT {cols_str} FROM {keyspace}.{table}"
37
38         if where_clause:
39             query += f" WHERE {where_clause} ALLOW FILTERING"
40
41         rows = self.session.execute(query)
42         return [row._asdict() for row in rows]

```

Listing A.4: Clase CassandraClient (cassandra_client.py)

A.3.2. Función de conversión de UUIDs

Los identificadores UUID de Cassandra deben convertirse a cadenas para ser compatibles con Redshift:

```

1 def convert_uuids_to_strings(df):
2     """
3     Convierte columnas UUID de un DataFrame a strings.
4     El driver de Cassandra retorna objetos UUID que necesitan
5     conversión para PostgreSQL/Redshift.
6     """
7     if df is None or df.empty:
8         return df
9
10    for col in df.columns:
11        if df[col].dtype == 'object' and len(df) > 0:
12            first_val = df[col].iloc[0]
13            if first_val is not None and 'uuid.UUID' in str(type(first_val)):

```

```

14         df[col] = df[col].astype(str)
15
16     return df

```

Listing A.5: Función `convert_uuids_to_strings`

A.3.3. Cliente Redshift

El módulo `redshift_client.py` maneja la conexión y carga de datos hacia Amazon Redshift Serverless:

```

1  class RedshiftClient:
2      """Cliente Redshift compatible con AWS Lambda y Secrets Manager."""
3
4      def __init__(self):
5          self.conn = None
6          self._connect()
7
8      def _connect(self):
9          """Se establece conexión con Redshift Serverless"""
10         host = os.environ.get('REDSHIFT_HOST', '')
11         database = os.environ.get('REDSHIFT_DATABASE', 'dev')
12         username = os.environ.get('REDSHIFT_USERNAME', '')
13         password = os.environ.get('REDSHIFT_PASSWORD', '')
14
15         self.conn = redshift_connector.connect(
16             host=host,
17             database=database,
18             port=5439,
19             user=username,
20             password=password
21         )
22
23     def load_dataframe(self, df: pd.DataFrame, table_name: str,
24                       schema: str = 'public',
25                       if_exists: str = 'append'):
26         """Carga de un DataFrame de pandas a una tabla de Redshift"""
27         if df.empty:
28             return 0
29

```

```

30 full_table = f"{schema}.{table_name}"
31
32 if if_exists == 'replace':
33     self.execute(f"TRUNCATE TABLE {full_table}")
34
35 columns = ', '.join(df.columns)
36 placeholders = ', '.join(['%s'] * len(df.columns))
37 insert_sql = f"INSERT INTO {full_table} ({columns}) VALUES ({placeholders})"
38
39 data = [tuple(row) for row in df.itertuples(index=False, name=None)]
40
41 self.execute_many(insert_sql, data)
42
43 return len(data)

```

Listing A.6: Clase RedshiftClient (redshift_client.py)

A.4. Configuración de Amazon Redshift

A.4.1. Entorno Redshift Serverless

Tabla A.1: Configuración de Redshift Serverless

Parámetro	Valor
Región AWS	us-east-1
Namespace	waki-bi-namespace
Workgroup	waki-bi-workgroup
Capacidad base (RPU)	8
Base de datos	prod
Esquema	dw_microcreditos
Puerto	5439
Endpoint	waki-bi-workgroup.834437190166.us-east-1. redshift-serverless.amazonaws.com

Nota. Elaboración propia.

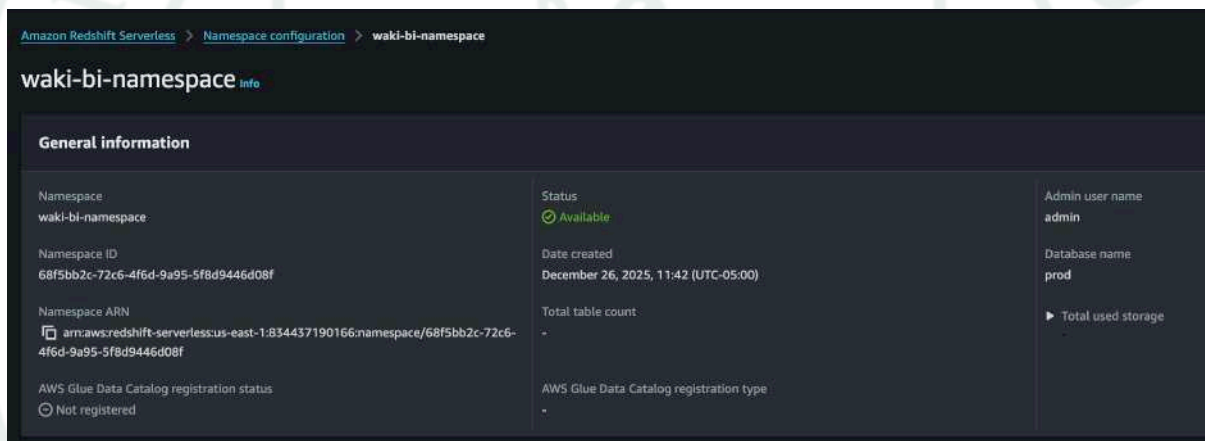
Las figuras A.1 y A.2 muestran la configuración del entorno Redshift Serverless en la consola de AWS.

Figura A.1: Configuración del Workgroup en Amazon Redshift Serverless



Nota. Elaboración propia.

Figura A.2: Configuración del Namespace en Amazon Redshift Serverless



Nota. Elaboración propia.

A.4.2. Resumen del modelo de datos

Tabla A.2: Resumen de tablas del datamart

Tabla	Tipo	Filas (aprox.)	Clave primaria
fact_cuota_microcredito	Hechos	6,293	sk_cuota
dim_cliente	Dimensión	2,874	sk_cliente
dim_credito	Dimensión	1,763	sk_credito
dim_tiempo	Dimensión	2,557	sk_tiempo
dim_producto	Dimensión	7	sk_producto
dim_estado_pago	Dimensión	8	sk_estado_pago
Total		13,502	

Nota. Elaboración propia.

La figura A.3 muestra una consulta ejecutada en el Redshift Query Editor con datos de ejemplo de la tabla de hechos.

Figura A.3: Consulta de datos en Redshift Query Editor

The screenshot shows a Redshift Query Editor window with the following SQL query: `SELECT * FROM "prod"."dw_microcreditos"."fact_cuota";`

The results are displayed in a table with the following columns: `sk_c...`, `sk_cliente`, `sk_credito`, `sk_tiempo_vencimiento`, `sk_tiempo_pago`, `sk_estado_pago`, `sk_producto`, and `numero_cuota`. The table contains 16 rows of data.

	sk_c...	sk_cliente	sk_credito	sk_tiempo_vencimiento	sk_tiempo_pago	sk_estado_pago	sk_producto	numero_cuota
1		4ecd3cb0-2905-11ea-8e...	81d86570-2b1d-11ea-a2...	20200131	0	4	3	1
2		4ecd3cb0-2905-11ea-8e...	81d86570-2b1d-11ea-a2...	20200229	0	4	3	2
3		4ecd3cb0-2905-11ea-8e...	81d86570-2b1d-11ea-a2...	20200331	0	4	3	3
4		8ded2d90-0a3f-11ea-a11...	518fcdad-b627-11ea-83f...	20200724	0	4	3	1
5		8ded2d90-0a3f-11ea-a11...	518fcdad-b627-11ea-83f...	20200824	0	4	3	2
6		8ded2d90-0a3f-11ea-a11...	518fcdad-b627-11ea-83f...	20200924	0	4	3	3
7		8ded2d90-0a3f-11ea-a11...	518fcdad-b627-11ea-83f...	20200917	0	1	3	0
8		fd1ab350-9edc-11e9-b41...	34a06080-90b8-11eb-bf3...	20210430	0	4	3	1
9		fd1ab350-9edc-11e9-b41...	34a06080-90b8-11eb-bf3...	20210530	0	4	3	2
10		fd1ab350-9edc-11e9-b41...	34a06080-90b8-11eb-bf3...	20210630	0	4	9	3
11		ca56bd02-7072-48a0-90...	e4ae0d90-df80-4995-8bd...	20250609	0	4	2	1
12		831e8857-2441-4abc-98...	157667a5-0c38-455e-8e...	20250901	0	4	1	1
13		831e8857-2441-4abc-98...	157667a5-0c38-455e-8e...	20251001	0	4	1	2
14		831e8857-2441-4abc-98...	157667a5-0c38-455e-8e...	20251101	0	4	1	3
15		831e8857-2441-4abc-98...	157667a5-0c38-455e-8e...	20251201	0	4	1	4
16		831e8857-2441-4abc-98...	157667a5-0c38-455e-8e...	20260101	0	1	1	5

Nota. Elaboración propia.

A.5. Definición de datos (DDL)

A.5.1. Dimensión: dim_cliente

```
1 CREATE TABLE dw_microcreditos.dim_cliente (  
2   sk_cliente bigint NOT NULL ENCODE az64,  
3   codigo_usuario character varying(50) ENCODE lzo,  
4   nombres character varying(200) ENCODE lzo,  
5   apellidos character varying(200) ENCODE lzo,  
6   dni character varying(20) ENCODE lzo,  
7   email character varying(200) ENCODE lzo,  
8   telefono character varying(50) ENCODE lzo,  
9   fecha_registro date ENCODE az64,  
10  tipo_cliente character varying(50) ENCODE lzo,  
11  estado_civil character varying(50) ENCODE lzo,  
12  genero character varying(10) ENCODE lzo,  
13  fecha_nacimiento date ENCODE az64,  
14  distrito character varying(100) ENCODE lzo,  
15  provincia character varying(100) ENCODE lzo,  
16  departamento character varying(100) ENCODE lzo,  
17  tipo_vivienda character varying(50) ENCODE lzo,  
18  ocupacion character varying(100) ENCODE lzo,  
19  empresa character varying(200) ENCODE lzo,  
20  salario_mensual numeric(18, 2) ENCODE az64,  
21  fecha_actualizacion timestamp ENCODE az64,  
22  PRIMARY KEY (sk_cliente)  
23 ) DISTSTYLE AUTO;
```

Listing A.7: DDL de la dimensión cliente

A.5.2. Dimensión: dim_estado_pago

```
1 CREATE TABLE dw_microcreditos.dim_estado_pago (  
2   sk_estado_pago integer NOT NULL ENCODE az64,  
3   estado_cuota character varying(50) ENCODE lzo,  
4   estado_pago character varying(50) ENCODE lzo,  
5   descripcion character varying(200) ENCODE lzo,  
6   PRIMARY KEY (sk_estado_pago)  
7 ) DISTSTYLE AUTO;
```

Listing A.8: DDL de la dimensión estado de pago

A.5.3. Dimensión: dim_tiempo

```
1 CREATE TABLE dw_microcreditos.dim_tiempo (  
2   sk_tiempo integer NOT NULL ENCODE az64,  
3   fecha date ENCODE az64,  
4   dia integer ENCODE az64,  
5   mes integer ENCODE az64,  
6   nombre_mes character varying(20) ENCODE lzo,  
7   trimestre integer ENCODE az64,  
8   anio integer ENCODE az64,  
9   dia_semana integer ENCODE az64,  
10  nombre_dia character varying(20) ENCODE lzo,  
11  es_fin_semana boolean ENCODE raw,  
12  semana_anio integer ENCODE az64,  
13  PRIMARY KEY (sk_tiempo)  
14 ) DISTSTYLE AUTO;
```

Listing A.9: DDL de la dimensión tiempo

A.5.4. Dimensión: dim_credito

```
1 CREATE TABLE dw_microcreditos.dim_credito (  
2   sk_credito bigint NOT NULL ENCODE az64,  
3   codigo_credito character varying(50) ENCODE lzo,  
4   monto_solicitado numeric(18, 2) ENCODE az64,  
5   monto_aprobado numeric(18, 2) ENCODE az64,  
6   tasa_interes numeric(8, 4) ENCODE az64,  
7   tasa_mora numeric(8, 4) ENCODE az64,  
8   plazo_meses integer ENCODE az64,  
9   cuota_mensual numeric(18, 2) ENCODE az64,  
10  tea numeric(8, 4) ENCODE az64,  
11  tcea numeric(8, 4) ENCODE az64,  
12  gastos_administrativos numeric(18, 2) ENCODE az64,  
13  seguro_desgravamen numeric(18, 2) ENCODE az64,  
14  fecha_solicitud date ENCODE az64,  
15  fecha_aprobacion date ENCODE az64,
```

```

16 fecha_desembolso date ENCODE az64,
17 estado_credito character varying(50) ENCODE lzo,
18 banco_desembolso character varying(200) ENCODE lzo,
19 score numeric(5, 2) ENCODE az64,
20 motivo_rechazo character varying(500) ENCODE lzo,
21 PRIMARY KEY (sk_credito)
22 ) DISTSTYLE AUTO;

```

Listing A.10: DDL de la dimensión crédito

A.5.5. Dimensión: dim_producto

```

1 CREATE TABLE dw_microcreditos.dim_producto (
2   sk_producto integer NOT NULL ENCODE az64,
3   tipo_producto character varying(50) ENCODE lzo,
4   subtipo_producto character varying(100) ENCODE lzo,
5   user_origin character varying(100) ENCODE lzo,
6   user_origin_app character varying(100) ENCODE lzo,
7   descripcion character varying(200) ENCODE lzo,
8   PRIMARY KEY (sk_producto)
9 ) DISTSTYLE AUTO;

```

Listing A.11: DDL de la dimensión producto

A.5.6. Tabla de hechos: fact_cuota_microcredito

```

1 CREATE TABLE dw_microcreditos.fact_cuota_microcredito (
2   sk_cuota bigint NOT NULL ENCODE az64,
3   sk_cliente bigint ENCODE az64,
4   sk_credito bigint ENCODE az64,
5   sk_tiempo_vencimiento integer ENCODE az64,
6   sk_tiempo_pago integer ENCODE az64,
7   sk_estado_pago integer ENCODE az64,
8   sk_producto integer ENCODE az64,
9   numero_cuota integer ENCODE az64,
10  total_cuotas integer ENCODE az64,
11  monto_cuota numeric(18, 2) ENCODE az64,
12  monto_capital numeric(18, 2) ENCODE az64,
13  monto_interes numeric(18, 2) ENCODE az64,

```

```

14 gastos_administrativos numeric(18, 2) ENCODE az64,
15 seguro_desgravamen numeric(18, 2) ENCODE az64,
16 igv numeric(18, 2) ENCODE az64,
17 monto_mora numeric(18, 2) ENCODE az64,
18 monto_pagado numeric(18, 2) ENCODE az64,
19 saldo_pendiente numeric(18, 2) ENCODE az64,
20 dias_mora integer ENCODE az64,
21 fecha_vencimiento date ENCODE az64,
22 fecha_creacion timestamp ENCODE az64,
23 fecha_actualizacion timestamp ENCODE az64,
24 estado_cuota character varying(50) ENCODE lzo,
25 estado_pago character varying(50) ENCODE lzo,
26 flag_vencida boolean ENCODE raw,
27 flag_pagada boolean ENCODE raw,
28 flag_con_mora boolean ENCODE raw,
29 PRIMARY KEY (sk_cuota)
30 ) DISTSTYLE AUTO;

```

Listing A.12: DDL de la tabla de hechos

A.6. Dicionarios de datos

A.6.1. Dimensión: dim_cliente

Tabla A.3: Diccionario de datos - dim_cliente (demografía)

Columna	Tipo	Nulo	Descripción
fecha_registro	DATE	Sí	Fecha de registro en la plataforma
tipo_cliente	VARCHAR(50)	Sí	Waki Planilla o Waki Consumo
estado_civil	VARCHAR(50)	Sí	Estado civil del cliente
genero	VARCHAR(10)	Sí	Género
fecha_nacimiento	DATE	Sí	Fecha de nacimiento
distrito	VARCHAR(100)	Sí	Distrito de residencia
provincia	VARCHAR(100)	Sí	Provincia de residencia
departamento	VARCHAR(100)	Sí	Departamento de residencia
tipo_vivienda	VARCHAR(50)	Sí	Tipo de vivienda

Nota. Elaboración propia.

Tabla A.4: Diccionario de datos - dim_cliente (información laboral)

Columna	Tipo	Nulo	Descripción
ocupacion	VARCHAR(100)	Sí	Ocupación o profesión
empresa	VARCHAR(200)	Sí	Nombre de la empresa empleadora
salario_mensual	NUMERIC(18,2)	Sí	Salario mensual declarado
fecha_actualizacion	TIMESTAMP	Sí	Fecha y hora de última actualización

Nota. Elaboración propia.

Tabla A.5: Diccionario de datos - dim_cliente (identificación)

Columna	Tipo	Nulo	Descripción
sk_cliente	BIGINT	No	Clave sustituta (PK)
codigo_usuario	VARCHAR(50)	Sí	UUID del usuario en sistema fuente
nombres	VARCHAR(200)	Sí	Nombres del cliente
apellidos	VARCHAR(200)	Sí	Apellidos del cliente
dni	VARCHAR(20)	Sí	Documento Nacional de Identidad
email	VARCHAR(200)	Sí	Correo electrónico registrado
telefono	VARCHAR(50)	Sí	Número de teléfono celular

Nota. Elaboración propia.

A.6.2. Dimensión: dim_credito

Tabla A.6: Diccionario de datos - dim_credito (identificación y montos)

Columna	Tipo	Nulo	Descripción
sk_credito	BIGINT	No	Clave sustituta (PK)
codigo_credito	VARCHAR(50)	Sí	UUID del microcrédito en sistema fuente
monto_solicitado	NUMERIC(18,2)	Sí	Monto solicitado por el cliente
monto_aprobado	NUMERIC(18,2)	Sí	Monto aprobado por Waki
plazo_meses	INTEGER	Sí	Plazo del crédito en meses
cuota_mensual	NUMERIC(18,2)	Sí	Monto estimado de cuota mensual

Nota. Elaboración propia.

Tabla A.7: Diccionario de datos - dim_credito (tasas y costos)

Columna	Tipo	Nulo	Descripción
tasa_interes	NUMERIC(8,4)	Sí	Tasa de interés mensual
tasa_mora	NUMERIC(8,4)	Sí	Tasa de mora diaria por atraso
tea	NUMERIC(8,4)	Sí	Tasa Efectiva Anual
tcea	NUMERIC(8,4)	Sí	Tasa de Costo Efectivo Anual
gastos_administrativos	NUMERIC(18,2)	Sí	Gastos administrativos
seguro_desgravamen	NUMERIC(18,2)	Sí	Prima de seguro de desgravamen

Nota. Elaboración propia.

Tabla A.8: Diccionario de datos - dim_credito (fechas y estado)

Columna	Tipo	Nulo	Descripción
fecha_solicitud	DATE	Sí	Fecha en que se solicitó el crédito
fecha_aprobacion	DATE	Sí	Fecha de aprobación del crédito
fecha_desembolso	DATE	Sí	Fecha de desembolso efectivo
estado_credito	VARCHAR(50)	Sí	Tipo: Activo, Retrasado, Finalizado
banco_desembolso	VARCHAR(200)	Sí	Banco destino del desembolso
score	NUMERIC(5,2)	Sí	Score crediticio del cliente
motivo_rechazo	VARCHAR(500)	Sí	Motivo de rechazo si fue rechazado

Nota. Elaboración propia.

A.6.3. Dimensión: dim_estado_pago

Tabla A.9: Diccionario de datos - dim_estado_pago

Columna	Tipo	Nulo	Descripción
sk_estado_pago	INTEGER	No	Clave sustituta (PK)
estado_cuota	VARCHAR(50)	Sí	Activo, Vencida, Finalizado
estado_pago	VARCHAR(50)	Sí	Pendiente, Aprobado
descripcion	VARCHAR(200)	Sí	Descripción del estado

Nota. Elaboración propia.

A.6.4. Dimensión: dim_producto

Tabla A.10: Diccionario de datos - dim_producto

Columna	Tipo	Nulo	Descripción
sk_producto	INTEGER	No	Clave sustituta (PK)
tipo_producto	VARCHAR(50)	Sí	Producto (Waki Planilla, Waki Consumo)
subtipo_producto	VARCHAR(100)	Sí	Subtipo específico del producto
user_origin	VARCHAR(100)	Sí	Canal de origen (Web, App)
user_origin_app	VARCHAR(100)	Sí	Origen en app (Simulador, Facebook)
descripcion	VARCHAR(200)	Sí	Descripción completa del producto

Nota. Elaboración propia.

A.6.5. Dimensión: dim_tiempo

Tabla A.11: Diccionario de datos - dim_tiempo

Columna	Tipo	Nulo	Descripción
sk_tiempo	INTEGER	No	Clave en formato YYYYMMDD
fecha	DATE	Sí	Fecha completa
dia	INTEGER	Sí	Día del mes (1-31)
mes	INTEGER	Sí	Número del mes (1-12)
nombre_mes	VARCHAR(20)	Sí	Nombre del mes
trimestre	INTEGER	Sí	Trimestre del año (1-4)
anio	INTEGER	Sí	Año (2019-2026)
dia_semana	INTEGER	Sí	(0=Lunes, 6=Domingo)
nombre_dia	VARCHAR(20)	Sí	Nombre del día
es_fin_semana	BOOLEAN	Sí	TRUE si es sábado o domingo
semana_anio	INTEGER	Sí	Número de semana del año (1-52)

Nota. Elaboración propia.

A.6.6. Tabla de hechos: fact_cuota_microcredito

Tabla A.12: Diccionario de datos - fact_cuota_microcredito (medidas)

Columna	Tipo	Nulo	Descripción
numero_cuota	INTEGER	Sí	Número secuencial de la cuota
total_cuotas	INTEGER	Sí	Número total de cuotas del crédito
monto_cuota	NUMERIC(18,2)	Sí	Monto total programado de la cuota
monto_capital	NUMERIC(18,2)	Sí	Componente de capital de la cuota
monto_interes	NUMERIC(18,2)	Sí	Componente de intereses de la cuota
gastos_administrativos	NUMERIC(18,2)	Sí	Gastos administrativos incluidos
seguro_desgravamen	NUMERIC(18,2)	Sí	Prima de seguro de desgravamen
igv	NUMERIC(18,2)	Sí	Impuesto General a las Ventas
monto_mora	NUMERIC(18,2)	Sí	Monto acumulado por mora
monto_pagado	NUMERIC(18,2)	Sí	Monto efectivamente pagado
saldo_pendiente	NUMERIC(18,2)	Sí	Saldo restante por pagar
dias_mora	INTEGER	Sí	Cantidad de días de atraso

Nota. Elaboración propia.

Tabla A.13: Diccionario de datos - fact_cuota_microcredito (claves)

Columna	Tipo	Nulo	Descripción
sk_cuota	BIGINT	No	Clave sustituta, identificador único de la cuota
sk_cliente	BIGINT	Sí	Clave foránea hacia dim_cliente
sk_credito	BIGINT	Sí	Clave foránea hacia dim_credito
sk_tiempo_vencimiento	INTEGER	Sí	Clave foránea a dim_tiempo
sk_tiempo_pago	INTEGER	Sí	Clave foránea a dim_tiempo
sk_estado_pago	INTEGER	Sí	Clave foránea a dim_estado_pago
sk_producto	INTEGER	Sí	Clave foránea hacia dim_producto

Nota. Elaboración propia.

Tabla A.14: Diccionario de datos - fact_cuota_microcredito (atributos generados)

Columna	Tipo	Nulo	Descripción
fecha_vencimiento	DATE	Sí	Fecha programada de vencimiento
fecha_creacion	TIMESTAMP	Sí	Fecha de creación del registro
fecha_actualizacion	TIMESTAMP	Sí	Fecha de última actualización
estado_cuota	VARCHAR(50)	Sí	Estado actual de la cuota
estado_pago	VARCHAR(50)	Sí	Estado actual del pago
flag_vencida	BOOLEAN	Sí	TRUE si la cuota está vencida
flag_pagada	BOOLEAN	Sí	TRUE si la cuota fue pagada
flag_con_mora	BOOLEAN	Sí	TRUE si monto_mora >0

Nota. Elaboración propia.

A.7. Contenido de dimensiones estáticas

A.7.1. Dimensión: dim_producto

La dimensión producto contiene las 7 categorías en las que Waki ofrece sus productos, clasificadas por línea de negocio y canal de adquisición, tal como se presenta en la Tabla A.15.

Tabla A.15: Valores de dim_producto

SK	Tipo	Subtipo	Origin	App	Descripción	
1	Waki Planilla	Crédito Convenio	por	NULL	NULL	Crédito con convenio empresarial
2	Waki Planilla	Adelanto Sueldo	de	NULL	NULL	Adelanto de sueldo vía planilla
3	Waki Consumo	Crédito Web	Web	NULL	NULL	Crédito desde portal web
4	Waki Consumo	Crédito App	App	NULL	NULL	Crédito desde aplicación móvil
5	Waki Consumo	Simulador Waki	NULL	Simulador	Simulador	Desde simulador en la app
6	Waki Consumo	Facebook	Web	Facebook	Facebook	Desde campaña de Facebook
7	Desconocido	Desconocido	NULL	NULL	NULL	Origen no identificado

Nota. Elaboración propia.

A.7.2. Dimensión: dim_estado_pago

La dimensión estado de pago define las 8 combinaciones posibles del estado de una cuota, descritas en la Tabla A.16:

Tabla A.16: Valores de dim_estado_pago

SK	Estado Cuota	Estado Pago	Descripción
1	Activo	Pendiente	Cuota activa, aún no vencida, sin pago registrado
2	Pendiente	Pendiente	Cuota en proceso de pago
3	Vencida	Pendiente	Cuota vencida sin pago (en mora)
4	Finalizado	Aprobado	Cuota pagada y cerrada correctamente
5	Activo	Aprobado	Cuota activa con pago anticipado registrado
6	Pendiente	Aprobado	Cuota pendiente con pago aprobado
7	Vencida	Aprobado	Cuota vencida con pago posterior (regularizada)
8	Finalizado	Pendiente	Cuota cerrada con pago pendiente de verificación

Nota. Elaboración propia.

A.8. Configuración de funciones Lambda

A.8.1. Inventario de funciones

Tabla A.17: Funciones Lambda y fuentes de datos

Función Lambda	Keyspace	Tablas extraídas
etl-dim-tiempo	(generación local)	N/A - genera fechas 2019-2026
etl-dim-estado-pago	(generación local)	N/A - 8 estados predefinidos
etl-dim-producto	(generación local)	N/A - 7 productos predefinidos
etl-dim-cliente	user_management_ms admission_ms	user user_employment payroll
etl-dim-credito	microcredit_mag_ms req_microcredit_ms	microcredits amount disbursement schedule
etl-fact-cuota	microcredit_mag_ms req_microcredit_ms	payment_schedule quota_details schedule microcredits

Nota. Elaboración propia.

A.8.2. Configuración técnica

Tabla A.18: Configuración técnica de funciones Lambda

Función	Memoria	Timeout	Imagen	Handler
etl-dim-tiempo	1024 MB	15 min	waki-etl:latest	handler.handler
etl-dim-estado-pago	1024 MB	15 min	waki-etl:latest	handler.handler
etl-dim-producto	1024 MB	15 min	waki-etl:latest	handler.handler
etl-dim-cliente	1024 MB	15 min	waki-etl:latest	handler.handler
etl-dim-credito	1024 MB	15 min	waki-etl:latest	handler.handler
etl-fact-cuota	1024 MB	15 min	waki-etl:latest	handler.handler

Nota. Elaboración propia.

A.8.3. Variables de entorno

Tabla A.19: Variables de entorno de las funciones Lambda

Variable	Descripción
CASSANDRA_CONTACT_POINTS	IP o hostname del cluster Cassandra
CASSANDRA_DATACENTER	Nombre del datacenter
CASSANDRA_USERNAME	Usuario de Cassandra
CASSANDRA_PASSWORD	Contraseña de Cassandra
REDSHIFT_HOST	Endpoint del workgroup Redshift Serverless
REDSHIFT_DATABASE	Nombre de la base de datos (prod)
REDSHIFT_USERNAME	Usuario administrador de Redshift
REDSHIFT_PASSWORD	Contraseña de Redshift
USER _MANAGEMENT _KEYSPACE	Keyspace de gestión de usuarios
MICROCREDIT _MANAGEMENT _KEYSPACE	Keyspace de gestión de créditos
REQUEST _MICROCREDIT _KEYSPACE	Keyspace de solicitudes de crédito
ADMISSION_KEYSPACE	Keyspace de admisión

Nota. Elaboración propia.

A.9. Dashboards en Power BI

A.9.1. Arquitectura de conexión híbrida

La conexión entre Power BI y Amazon Redshift Serverless requiere una arquitectura híbrida debido a que el clúster de Redshift se encuentra en una VPC privada sin acceso público directo. Se implementaron dos flujos de trabajo diferenciados:

- **Entorno de Desarrollo:** Conexión desde Power BI Desktop mediante túnel SSM y bridge TCP.
- **Entorno de Producción:** Conexión desde Power BI Service mediante On-premises Data Gateway.

A.9.1.1. Componentes de la arquitectura

Tabla A.20: Componentes de la arquitectura de conexión

Componente	Entorno	Función
Amazon Redshift Serverless	AWS (VPC privada)	Data Warehouse con el esquema dw_microcreditos
Instancia EC2 Windows	AWS (VPC privada)	Servidor Gateway/Bastión
On-premises Data Gateway	Instalado en EC2	Conexión segura para Power BI Service
AWS SSM Port Forwarding	Túnel encriptado	Canal seguro desde estación local
Power BI Desktop	Local (desarrollo)	Diseño y desarrollo de reportes
Power BI Service	Nube (producción)	Publicación y actualización automática

Nota. Elaboración propia.

A.9.2. Conexión con Redshift

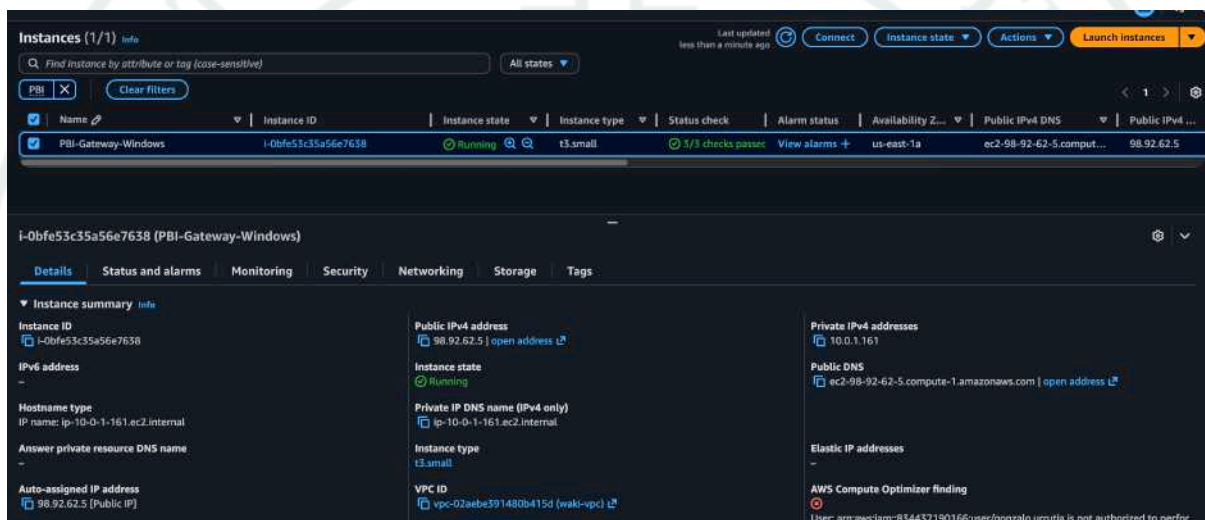
A.9.2.1. Flujo de desarrollo con Power BI Desktop

El proceso de conexión desde el entorno de desarrollo requiere cuatro pasos secuenciales:

Paso 1: Iniciar infraestructura EC2

1. Acceder a la consola de AWS EC2
2. Seleccionar la instancia PBI-Gateway-Windows (ID: i-0bfe53c35a56e7638)
3. Ejecutar **Instance State** > **Start Instance**
4. Verificar que el estado cambie a **Running** y los Status Checks estén en verde, como se muestra en la Figura A.4

Figura A.4: Consola de AWS EC2 mostrando la instancia PBI-Gateway-Windows en estado Running



Nota. Elaboración propia.

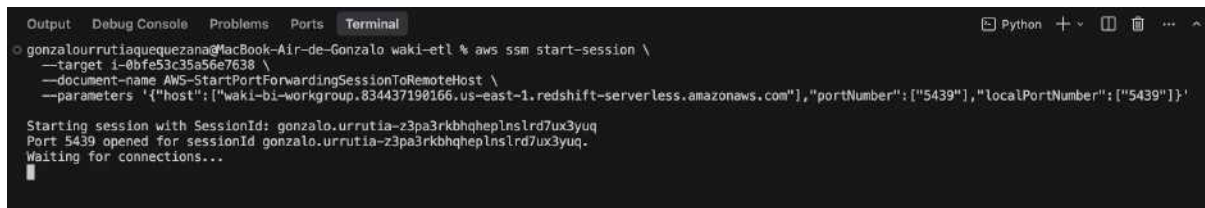
Paso 2: Establecer túnel SSM Se crea un canal encriptado desde el puerto local 5439 hacia el endpoint de Redshift:

```
1 aws ssm start-session \  
2 --target i-0bfe53c35a56e7638 \  
3 --document-name AWS-StartPortForwardingSessionToRemoteHost \  
4 --parameters '{"host":["waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.  
  ↪ amazonaws.com"],"portNumber":["5439"],"localPortNumber":["5439"]}'
```

Listing A.13: Comando para establecer túnel SSM desde macOS

Asimismo en la Figura A.5 se muestra la ejecución del comando el cual debe permanecer en ejecución, el mensaje Waitingforconnections indica que el túnel está activo.

Figura A.5: Terminal mostrando el túnel SSM activo con mensaje “Waiting for connections”



```
Output Debug Console Problems Ports Terminal
gonzalourrutiaquequezana@MacBook-Air-de-Gonzalo waki-eti % aws ssm start-session \
--target i-0bfe53c35a56e7638 \
--document-name AWS-StartPortForwardingSessionToRemoteHost \
--parameters '{"host":["waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com"],"portNumber":["5439"],"localPortNumber":["5439"]}'

Starting session with SessionId: gonzalo.urrutia-z3pa3rkbhqheplnsld7ux3yuq
Port 5439 opened for sessionId gonzalo.urrutia-z3pa3rkbhqheplnsld7ux3yuq.
Waiting for connections...
```

Nota. Elaboración propia.

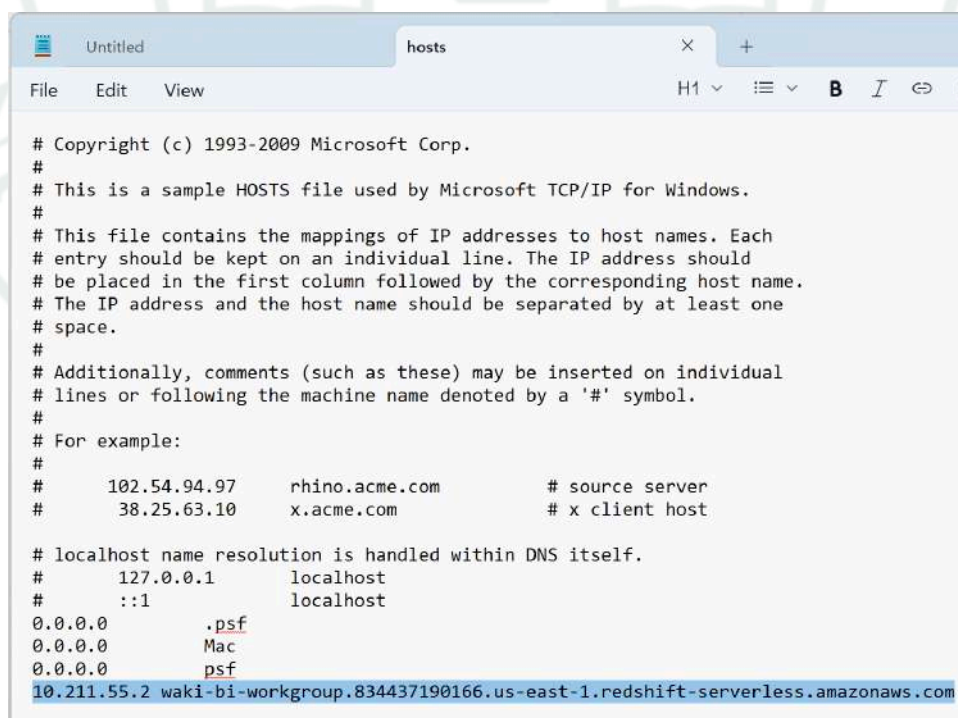
Paso 3: Configurar bridge de red Debido a que el túnel SSM solo escucha en localhost y no es accesible desde máquinas virtuales, se utiliza socat para exponer el puerto a la interfaz de red compartida, este comando redirige el tráfico del puerto 5440 (accesible desde Parallels/-Windows VM) al puerto 5439 (túnel AWS).

```
1 socat -d -d TCP-LISTEN:5440,bind=0.0.0.0,fork TCP:127.0.0.1:5439
```

Listing A.14: Comando para bridge TCP con socat

Paso 4: Configurar DNS Spoofing en Windows Para evitar errores de validación de certificado SSL, se añade la entrada correspondiente en el archivo C:\Windows\System32\drivers\etc\hosts, tal como se muestra en la Figura A.6, esto fuerza a Windows a resolver el dominio AWS hacia la IP local del host macOS (gateway de Parallels).

Figura A.6: Archivo hosts de Windows mostrando la entrada DNS personalizada



```
Untitled hosts
File Edit View H1 B I
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97    rhino.acme.com          # source server
#       38.25.63.10   x.acme.com              # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1     localhost
#       ::1           localhost
0.0.0.0       .psf
0.0.0.0       Mac
0.0.0.0       psf
10.211.55.2   waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com
```

Nota. Elaboración propia.

A.9.2.2. Conexión desde Power BI Desktop

Una vez establecida la infraestructura de túnel, la conexión en Power BI Desktop se configura con los parámetros indicados en la Tabla A.21. El diálogo de conexión resultante se muestra en la Figura A.7, y las 6 tablas del esquema dw_microcreditos cargadas en el navegador de Power BI se presentan en la Figura A.8.

Tabla A.21: Parámetros de conexión Power BI Desktop

Parámetro	Valor
Tipo de origen	Amazon Redshift
Servidor	waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com:5440
Base de datos	prod
Modo de datos	DirectQuery (recomendado para desarrollo)
Autenticación	Database (usuario y contraseña)

Nota. Elaboración propia.

Figura A.7: Diálogo de conexión de Power BI Desktop con los parámetros configurados

Amazon Redshift

Server
waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com:5440

Database
prod

▲ Opciones avanzadas
Especifique un valor de texto para usarlo como nombre de proveedor (opcional)
Ejemplo: Provider Name

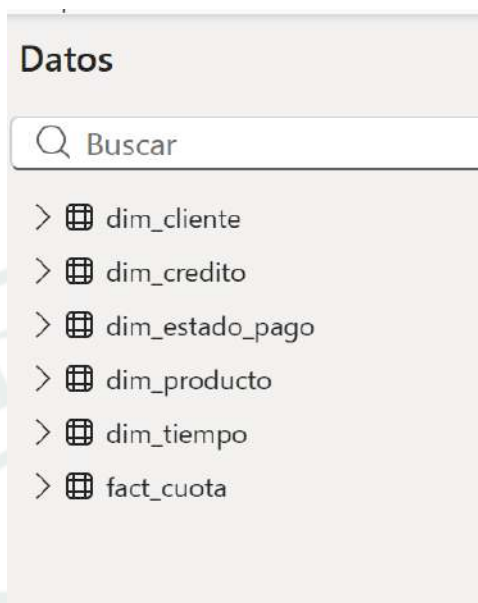
Tamaño del lote
Ejemplo: 100

Instrucción SQL (opcional, requiere una base de datos)
Ejemplo: seLe...

Aceptar Cancelar

Nota. Elaboración propia.

Figura A.8: Tablas del esquema dw_microcreditos en Power BI



Nota. Elaboración propia.

A.9.2.3. Flujo de producción con Power BI Service

Para la actualización automática de reportes publicados en Power BI Service, se utiliza el **On-premises Data Gateway** instalado en la instancia EC2, cuya configuración se detalla en la Tabla A.22. El estado en línea del gateway puede verificarse en la consola de Power BI Service, tal como se ilustra en la Figura A.9.

Tabla A.22: Configuración del On-premises Data Gateway

Parámetro	Valor
Servicio	Microsoft On-premises Data Gateway
Nombre del Gateway	Gateway-Redshift-Prod
Estado	Online (cuando EC2 está encendida)
Tipo de conexión	Local (On-premises)
Servidor Redshift	waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.amazonaws.com:5439
Autenticación	Basic (Database credentials)

Nota. Elaboración propia.

Figura A.9: Consola de Power BI Service mostrando el Gateway en estado Online



Nota. Elaboración propia.

A.9.2.4. Configuración de Security Groups en AWS

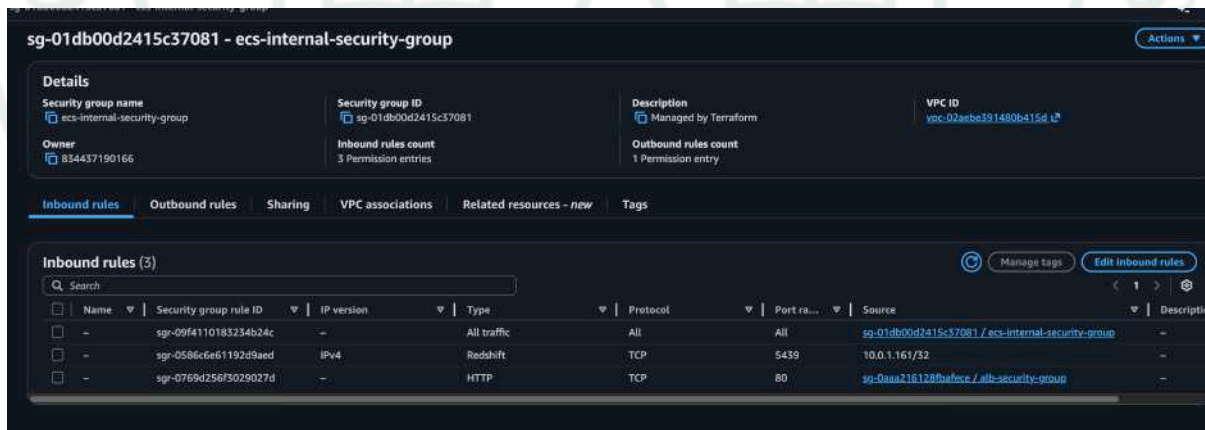
Para permitir la comunicación entre el Gateway y Redshift, se configuró una regla de entrada en el Security Group de Redshift, especificada en la Tabla A.23 y visible en la consola de AWS en la Figura A.10.

Tabla A.23: Regla de Security Group para Redshift

Tipo	Puerto	Origen	Descripción
Custom TCP	5439	10.0.1.161/32	IP privada de EC2 Gateway

Nota. Elaboración propia.

Figura A.10: AWS Security Group mostrando la regla de entrada configurada para Redshift



Nota. Elaboración propia.

A.9.2.5. Conexión en Power BI Service

Para configurar la conexión desde Power BI Service, tal como se ilustra en la Figura A.11, se deben seguir los siguientes pasos:

1. Acceder a **Configuración > Administrar conexiones y puertas de enlace**

2. Crear nueva conexión de tipo **On-premises**
3. Seleccionar el cluster **Gateway-Redshift-Prod**
4. Configurar el endpoint original de Redshift (puerto 5439)
5. Ingresar credenciales de autenticación básica

Figura A.11: Configuración de conexión en Power BI Service con el Gateway

Configuración ✕

Nombre de cluster de puerta de enlace

Gateway-Redshift-Prod

Nombre de conexión *

Conexion-Redshift-Waki

Id. de conexión

4885e27e-0829-4c1c-aa1a-062a6dd1330a

Tipo de conexión

Amazon Redshift

Ruta de acceso del origen de datos

waki-bi-workgroup.834437190166.us-east-1.redshift-serverless.a...

Autenticación

Método de autenticación *

Básica

Nombre de usuario *

admin

Contraseña *

.....

Omitir conexión de prueba

Las credenciales se cifran mediante la clave almacenada en el entorno local del servidor de puerta de enlace. [Para obtener más información, vea esta información general.](#)

Nota. Elaboración propia.

A.9.3. Consideraciones operativas

A.9.3.1. Apagado de recursos

Para optimizar costos, la instancia EC2 debe apagarse al finalizar la jornada de trabajo. Cuando la instancia está apagada, el Gateway aparecerá como **Offline** en Power BI Service, pero se reconectará automáticamente al encender la instancia nuevamente.

A.9.3.2. Solución de problemas comunes

Tabla A.24: Errores comunes y soluciones

Error	Solución
host name must be specified for a verified SSL connection	Verificar que el archivo hosts de Windows contenga la entrada DNS correcta
Unable to connect: Connection refused	Verificar que los dos túneles (SSM y socat) estén activos en las terminales
Gateway offline en Power BI Service	Iniciar la instancia EC2 desde la consola de AWS
Timeout connecting to Redshift	Verificar reglas del Security Group y que la IP privada de EC2 esté autorizada

Nota. Elaboración propia.

A.9.4. Medidas DAX implementadas

Las medidas DAX constituyen la lógica de cálculo del sistema de dashboards. Se implementaron un total de 31 medidas organizadas por página y funcionalidad. La tabla A.25 presenta el resumen por página y el código fuente completo de todas las medidas se presenta a continuación, organizado por categoría.

Tabla A.25: Resumen de medidas DAX por página

Página / Categoría	Cantidad
Medidas Auxiliares (Base)	6
Página 1: Gerencial - Empresas	5
Página 2: Gerencial - Empresas Detalle	4
Página 3: Operativo - Cobranza	10
Página 4: Operativo - Alertas y Riesgo	5
Tabla Auxiliar (Rangos Mora)	1
Total	31

Nota. Elaboración propia.

A.9.4.1. Medidas auxiliares (base)

```

1 // Saldo total del producto Waki Planilla
2 Saldo Waki Planilla =
3 CALCULATE(
4     SUM(fact_cuota[saldo_pendiente]),
5     dim_producto[tipo_producto] = "Waki Planilla"
6 )
7
8 // Saldo total de toda la cartera
9 Saldo Total Cartera =
10 SUM(fact_cuota[saldo_pendiente])
11
12 // Conteo de usuarios que tienen al menos un crédito
13 Usuarios con Credito =
14 CALCULATE(
15     DISTINCTCOUNT(fact_cuota[sk_cliente]),
16     dim_producto[tipo_producto] = "Waki Planilla"
17 )
18
19 // Usuarios con dos o más créditos (recurrentes)
20 Usuarios Recurrentes =
21 CALCULATE(
22     DISTINCTCOUNT(fact_cuota[sk_cliente]),
23     FILTER(
24         VALUES(fact_cuota[sk_cliente]),

```

```

25     CALCULATE(DISTINCTCOUNT(fact_cuota[sk_credito])) >= 2
26 ),
27     dim_producto[tipo_producto] = "Waki Planilla"
28 )
29
30 // Conteo total de microcréditos
31 Total Microcreditos =
32 CALCULATE(
33     DISTINCTCOUNT(fact_cuota[sk_credito]),
34     dim_producto[tipo_producto] = "Waki Planilla"
35 )
36
37 // Suma de montos en mora
38 Monto en Mora =
39 CALCULATE(
40     SUM(fact_cuota[monto_mora]),
41     fact_cuota[monto_mora] > 0
42 )

```

Listing A.15: Medidas DAX auxiliares

A.9.4.2. Medidas de la Página 1: Reporte Gerencial - Empresas

```

1 // Saldo de cartera con créditos activos o retrasados
2 Saldo Cartera Vigente =
3 CALCULATE(
4     [Saldo Waki Planilla],
5     dim_credito[estado_credito] IN {"Activo", "Retrasado"}
6 )
7
8 // Conteo de clientes tipo Waki Planilla
9 Total Wakitarios =
10 CALCULATE(
11     DISTINCTCOUNT(dim_cliente[sk_cliente]),
12     dim_cliente[tipo_cliente] = "Waki Planilla"
13 )
14
15 // Créditos en estado activo o retrasado
16 Creditos Vigentes =

```

```

17 CALCULATE(
18     [Total Microcreditos],
19     dim_credito[estado_credito] IN {"Activo", "Retrasado"}
20 )
21
22 // Empresas con al menos un crédito activo
23 Empresas Activas =
24 CALCULATE(
25     DISTINCTCOUNT(dim_cliente[empresa]),
26     dim_cliente[tipo_cliente] = "Waki Planilla",
27     dim_credito[estado_credito] = "Activo"
28 )
29
30 // Porcentaje de mora sobre cartera total
31 Tasa Morosidad =
32 DIVIDE([Monto en Mora], [Saldo Total Cartera], 0)

```

Listing A.16: Medidas DAX - Página 1: Gerencial Empresas

A.9.4.3. Medidas de la Página 2: Reporte Gerencial - Empresas Detalle

```

1 // Texto con cantidad y porcentaje de usuarios que han solicitado
2 Usuarios Han Solicitado =
3 VAR TotalConCredito = [Usuarios con Credito]
4 VAR TotalWakitarios = [Total Wakitarios]
5 RETURN
6 TotalConCredito & " (" &
7 FORMAT(DIVIDE(TotalConCredito, TotalWakitarios, 0), "0%") & ")"
8
9
10 // Texto con cantidad y porcentaje de usuarios recurrentes
11 Usuarios Recurrentes Texto =
12 VAR Recurrentes = [Usuarios Recurrentes]
13 VAR TotalConCredito = [Usuarios con Credito]
14 RETURN
15 Recurrentes & " (" &
16 FORMAT(DIVIDE(Recurrentes, TotalConCredito, 0), "0%") & ")"
17
18

```

```

19 // Nuevos wakitarios registrados en el mes seleccionado
20 Nuevos Wakitarios Mes =
21 VAR MesSeleccionado = SELECTEDVALUE(dim_tiempo[mes])
22 VAR AnioSeleccionado = SELECTEDVALUE(dim_tiempo[anio])
23 RETURN
24 CALCULATE(
25     DISTINCTCOUNT(dim_cliente[codigo_usuario]),
26     dim_cliente[tipo_cliente] = "Waki Planilla",
27     MONTH(dim_cliente[fecha_registro]) = MesSeleccionado,
28     YEAR(dim_cliente[fecha_registro]) = AnioSeleccionado
29 )
30
31
32 // Nuevos créditos desembolsados en el mes seleccionado
33 Nuevos Creditos Mes =
34 CALCULATE(
35     DISTINCTCOUNT(dim_credito[codigo_credito]),
36     dim_producto[tipo_producto] = "Waki Planilla",
37     FILTER(
38         ALL(dim_credito),
39         MONTH(dim_credito[fecha_desembolso]) =
40             SELECTEDVALUE(dim_tiempo[mes]) &&
41             YEAR(dim_credito[fecha_desembolso]) =
42             SELECTEDVALUE(dim_tiempo[anio])
43     )
44 )

```

Listing A.17: Medidas DAX - Página 2: Gerencial Detalle

A.9.4.4. Medidas de la Página 3: Reporte Operativo - Cobranza

```

1 // Cantidad de cuotas vencidas pendientes de pago
2 Cuotas Vencidas =
3 CALCULATE(
4     COUNTROWS(fact_cuota),
5     fact_cuota[flag_vencida] = TRUE,
6     fact_cuota[flag_pagada] = FALSE,
7     dim_producto[tipo_producto] = "Waki Planilla"
8 )

```

```

9
10 // Cuotas que vencerán en los próximos 7 días
11 Cuotas Por Vencer 7D =
12 VAR FechaHoy = TODAY()
13 VAR Fecha7Dias = TODAY() + 7
14 RETURN
15 CALCULATE(
16     COUNTROWS(fact_cuota),
17     fact_cuota[flag_pagada] = FALSE,
18     fact_cuota[fecha_vencimiento] >= FechaHoy,
19     fact_cuota[fecha_vencimiento] <= Fecha7Dias,
20     dim_producto[tipo_producto] = "Waki Planilla"
21 )
22
23 // Clientes con más de 15 días de mora
24 Clientes Mora 15D =
25 CALCULATE(
26     DISTINCTCOUNT(fact_cuota[sk_cliente]),
27     fact_cuota[dias_mora] > 15,
28     fact_cuota[flag_pagada] = FALSE,
29     dim_producto[tipo_producto] = "Waki Planilla"
30 )
31
32 // Monto total de cuotas vencidas pendientes
33 Monto Cuotas Vencidas =
34 CALCULATE(
35     SUM(fact_cuota[monto_cuota]),
36     fact_cuota[flag_vencida] = TRUE,
37     fact_cuota[flag_pagada] = FALSE,
38     dim_producto[tipo_producto] = "Waki Planilla"
39 )

```

Listing A.18: Medidas DAX - Página 3: Operativo Cobranza

```

1 // Clientes con 0 a 7 días de mora
2 Clientes 0-7 dias =
3 CALCULATE(
4     DISTINCTCOUNT(fact_cuota[sk_cliente]),
5     fact_cuota[dias_mora] >= 0,
6     fact_cuota[dias_mora] <= 7,

```

```

7 fact_cuota[flag_pagada] = FALSE,
8 dim_producto[tipo_producto] = "Waki Planilla"
9 )
10
11 // Clientes con 8 a 15 días de mora
12 Clientes 8–15 días =
13 CALCULATE(
14     DISTINCTCOUNT(fact_cuota[sk_cliente]),
15     fact_cuota[dias_mora] >= 8,
16     fact_cuota[dias_mora] <= 15,
17     fact_cuota[flag_pagada] = FALSE,
18     dim_producto[tipo_producto] = "Waki Planilla"
19 )
20
21 // Clientes con 16 a 30 días de mora
22 Clientes 16–30 días =
23 CALCULATE(
24     DISTINCTCOUNT(fact_cuota[sk_cliente]),
25     fact_cuota[dias_mora] >= 16,
26     fact_cuota[dias_mora] <= 30,
27     fact_cuota[flag_pagada] = FALSE,
28     dim_producto[tipo_producto] = "Waki Planilla"
29 )
30
31 // Clientes con 31 a 60 días de mora
32 Clientes 31–60 días =
33 CALCULATE(
34     DISTINCTCOUNT(fact_cuota[sk_cliente]),
35     fact_cuota[dias_mora] >= 31,
36     fact_cuota[dias_mora] <= 60,
37     fact_cuota[flag_pagada] = FALSE,
38     dim_producto[tipo_producto] = "Waki Planilla"
39 )
40
41 // Clientes con más de 60 días de mora
42 Clientes 60+ días =
43 CALCULATE(
44     DISTINCTCOUNT(fact_cuota[sk_cliente]),
45     fact_cuota[dias_mora] > 60,

```

```

46 fact_cuota[flag_pagada] = FALSE,
47 dim_producto[tipo_producto] = "Waki Planilla"
48 )
49
50 // Medida dinámica para gráfico de distribución
51 Clientes por Rango =
52 SWITCH(
53     SELECTEDVALUE('Rangos Mora'[Rango]),
54     "0-7 días", [Clientes 0-7 días],
55     "8-15 días", [Clientes 8-15 días],
56     "16-30 días", [Clientes 16-30 días],
57     "31-60 días", [Clientes 31-60 días],
58     "+60 días", [Clientes 60+ días],
59     BLANK()
60 )

```

Listing A.19: Medidas DAX - Página 3: Rangos de mora

A.9.4.5. Medidas de la Página 4: Reporte Operativo - Alertas y Riesgo

```

1 // Saldo de créditos con más de 30 días de mora
2 Cartera Riesgo 30D =
3 CALCULATE(
4     SUM(fact_cuota[saldo_pendiente]),
5     fact_cuota[dias_mora] > 30,
6     fact_cuota[flag_pagada] = FALSE,
7     dim_producto[tipo_producto] = "Waki Planilla"
8 )
9
10 // Clientes con más de 60 días de mora (críticos)
11 Clientes Criticos 60D =
12 CALCULATE(
13     DISTINCTCOUNT(fact_cuota[sk_cliente]),
14     fact_cuota[dias_mora] > 60,
15     fact_cuota[flag_pagada] = FALSE,
16     dim_producto[tipo_producto] = "Waki Planilla"
17 )
18
19 // Promedio de días de mora de cuotas atrasadas

```

```

20 Dias Mora Promedio =
21 CALCULATE(
22     AVERAGE(fact_cuota[dias_mora]),
23     fact_cuota[dias_mora] > 0,
24     dim_producto[tipo_producto] = "Waki Planilla"
25 )
26
27 // Clientes sin días de mora
28 Clientes Sin Mora =
29 CALCULATE(
30     DISTINCTCOUNT(fact_cuota[sk_cliente]),
31     fact_cuota[dias_mora] = 0,
32     dim_producto[tipo_producto] = "Waki Planilla"
33 )
34
35 // Clientes con más de 30 días de mora
36 Clientes 30+ días =
37 CALCULATE(
38     DISTINCTCOUNT(fact_cuota[sk_cliente]),
39     fact_cuota[dias_mora] > 30,
40     fact_cuota[flag_pagada] = FALSE,
41     dim_producto[tipo_producto] = "Waki Planilla"
42 )

```

Listing A.20: Medidas DAX - Página 4: Alertas y Riesgo

A.9.4.6. Tabla auxiliar para rangos de mora

```
1 Rangos Mora =  
2 DATATABLE(  
3   "Rango", STRING,  
4   "Orden", INTEGER,  
5   {  
6     {"0-7 días", 1},  
7     {"8-15 días", 2},  
8     {"16-30 días", 3},  
9     {"31-60 días", 4},  
10    {"+60 días", 5}  
11  }
```

Listing A.21: Tabla auxiliar para rangos de mora

A.9.5. Publicación y despliegue de dashboards

Para garantizar la integridad de los datos y evitar conflictos de versiones, se estableció una política estricta de flujo unidireccional donde el archivo local (.pbix) actúa como la fuente única de verdad. El flujo de trabajo implementado es:

Entorno Local (Power BI Desktop) → Publicación → Entorno Web (Power BI Service)

Las buenas prácticas implementadas incluyen:

- **Edición exclusiva en Desktop:** Todo cambio lógico (medidas DAX, relaciones, Power Query) o visual debe realizarse exclusivamente en Power BI Desktop.
- **Prohibición de edición web:** Se restringe la edición directa en el navegador para evitar conflictos que no puedan ser recuperados al entorno local.

A.9.5.1. Lógica de conexión: desarrollo vs. producción

El sistema opera bajo dos contextos de red distintos que requieren estandarización antes del despliegue:

- **Entorno de desarrollo (local):** Utiliza túnel SSM y socat a través del puerto 5440 para conectar Power BI Desktop con la VPC privada de AWS.

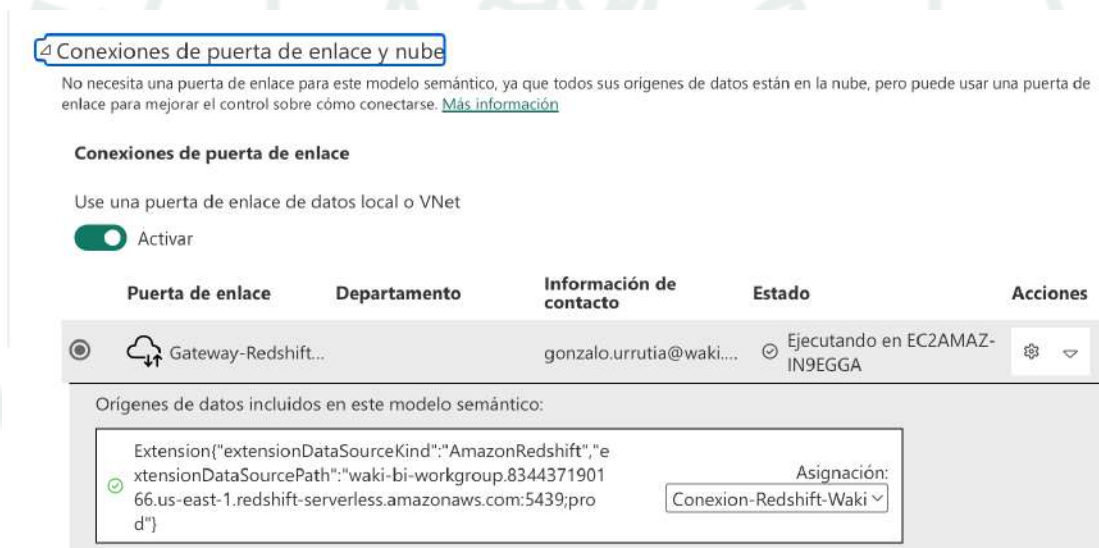
- **Entorno de producción (nube):** Utiliza el On-premises Data Gateway instalado en la instancia EC2 para conectar Power BI Service directamente con Redshift en el puerto 5439.

Paso crítico de pre-publicación Antes de publicar el archivo a producción, se debe asegurar que la cadena de conexión en el archivo .pbix apunte al puerto nativo de Redshift (:5439). Esto permite que, al llegar a la nube, el Gateway reconozca la solicitud y la enrute correctamente.

A.9.5.2. Configuración de puerta de enlace

Una vez publicado el reporte en Power BI Service, el Dataset (Modelo Semántico) queda inicialmente desconectado, para habilitar la actualización de datos, se debe realizar el mapeo lógico entre el origen de datos del archivo y la conexión física configurada en el Gateway. La figura A.12 muestra la configuración de mapeo en el entorno productivo:

Figura A.12: Configuración de mapeo de Gateway para publicación de dashboard



Nota. Elaboración propia.

Los pasos de configuración son:

1. **Gateway:** Se selecciona el clúster Gateway-Redshift-Prod alojado en la instancia EC2.
2. **Mapeo:** Se vincula la cadena de conexión detectada en el código del reporte con la credencial almacenada en el Gateway (Conexion-Redshift-Waki).
3. **Resultado:** El tráfico de actualización fluye de manera segura desde la nube de Microsoft hacia la subred privada de AWS sin exponer la base de datos a internet público.

A.9.5.3. Persistencia de datos

Se implementó una estrategia de costos eficiente aprovechando el modo de almacenamiento **Import (Importación)** de Power BI:

- **Visualización (lectura):** NO requiere que la instancia EC2 esté encendida. Al usar el modo Import, los datos residen en la memoria de la nube de Microsoft, los usuarios finales pueden consultar, filtrar e interactuar con el dashboard 24/7 sin generar costos de cómputo en AWS.
- **Actualización (refresh):** SÍ requiere la instancia EC2. La instancia actúa solo como "puente" durante el momento de la ingesta de datos nuevos.

La rutina de actualización implementada es:

1. Encendido de instancia EC2 (StartInstance)
2. Ejecución de actualización programada o manual en Power BI Service
3. Apagado de instancia EC2 (StopInstance) tras finalizar la carga

Esta lógica desacopla la disponibilidad del reporte de la infraestructura subyacente, garantizando alta disponibilidad a bajo costo.

A.10. Comandos de operación

A.10.1. Ejecución manual de funciones ETL

```
1 # Invocar etl-fact-cuota manualmente
2 aws lambda invoke \
3   --function-name etl-fact-cuota \
4   --payload '{}' \
5   --region us-east-1 \
6   response.json
7
8 # Verificar resultado
9 cat response.json
```

Listing A.22: Invocación manual de función Lambda

A.10.2. Consulta de logs en CloudWatch

```
1 # Ver logs de la ultima hora
2 aws logs tail /aws/lambda/etl-fact-cuota \
3 --since 1h \
4 --region us-east-1
```

Listing A.23: Consulta de logs recientes

A.11. Resumen del anexo

Este anexo documenta la implementación técnica completa de la solución:

- **Estructura del proyecto:** Organización de directorios, dependencias Python y Dockerfile para despliegue.
- **Módulos compartidos:** Código fuente de los clientes Cassandra y Redshift.
- **DDL completo:** Definición de las 6 tablas con tipos de datos, codificación y claves.
- **Diccionarios de datos:** Documentación de las 57 columnas distribuidas en 15 tablas de metadatos.
- **Dimensiones estáticas:** Valores de dim_producto (7 registros) y dim_estado_pago (8 registros).
- **Configuración Lambda:** 6 funciones con memoria, timeout, imágenes y 12 variables de entorno.
- **Dashboards Power BI:** Sistema de 4 páginas (2 gerenciales y 2 operativas) conectadas vía DirectQuery a Redshift, con medidas DAX para indicadores de cartera, morosidad y cobranza.

Anexo B: Acta de aprobación de KPIs

El presente anexo contiene el acta firmada por la administradora de Holding Waki SAC, mediante la cual se declara la conformidad con los indicadores clave de desempeño (KPIs) implementados en los cuatro dashboards de Power BI que conforman la solución de inteligencia de negocios.

Figura B.1: Acta de aprobación de KPIs firmada por la administradora — Página 1

HOLDING WAKI S.A.C.

Arequipa, Perú

ACTA DE APROBACIÓN DE INDICADORES CLAVE DE DESEMPEÑO (KPIs)

En la ciudad de Arequipa, la administradora de Holding Waki S.A.C., identificada al pie del presente documento, declara haber revisado y aprobado los indicadores clave de desempeño (KPIs) definidos e implementados en la solución de inteligencia de negocios desarrollada por el bachiller **Gonzalo Urrutia Quequezana**, como parte de su tesis titulada: *"Implementación de una solución de inteligencia de negocios para optimizar la generación de reportes de seguimiento de microcréditos en una empresa financiera en la ciudad de Arequipa"*.

Los KPIs aprobados corresponden a los cuatro dashboards implementados en Microsoft Power BI y se detallan a continuación:

Dashboard	KPIs aprobados
Reporte Gerencial — Empresas	<ul style="list-style-type: none">• Saldo de Cartera Vigente• Total de Wakitarios• Créditos Vigentes• Empresas Activas• Top 10 de empresas por saldo de cartera• Ranking de empresas con tasa de morosidad
Reporte Gerencial — Empresas Detalle	<ul style="list-style-type: none">• Total de Wakitarios de la empresa• Usuarios que han solicitado crédito• Créditos Vigentes• Usuarios Recurrentes• Evolución mensual de nuevos wakitarios• Tendencia de nuevos créditos por mes

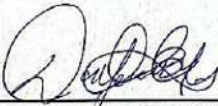
Nota. Elaboración propia.

Figura B.2: Acta de aprobación de KPIs firmada por la administradora — Página 2

Reporte Operativo — Cobranza	<ul style="list-style-type: none"> • Cuotas Vencidas • Cuotas por Vencer en 7 días • Clientes con más de 15 días de mora • Monto Total de Cuotas Vencidas • Distribución de clientes por rangos de días de mora
Reporte Operativo — Alertas y Riesgo	<ul style="list-style-type: none"> • Cartera en Riesgo • Tasa de Morosidad global • Clientes Críticos (más de 60 días de mora) • Promedio de Días de Mora • Matriz de riesgo por empresa • Evolución de la tasa de morosidad mensual

Los indicadores mencionados corresponden efectivamente a la información que la administración de Holding Waki S.A.C. requiere para el seguimiento operativo y gerencial de la cartera de microcréditos, y sustituyen adecuadamente los reportes manuales que se elaboraban previamente en hojas de cálculo.

En señal de conformidad, se firma la presente acta.



Dayana Berrio Vilella
 Administradora
 Holding Waki S.A.C.
 DNI: 47592676

Nota. Elaboración propia.

Anexo C: Acta de aprobación del diseño visual de dashboards

El presente anexo contiene el acta firmada por la administradora de Holding Waki SAC, mediante la cual se declara la conformidad con el diseño visual, la distribución de los elementos, los gráficos y el formato general de los cuatro dashboards implementados en Power BI. El acta incluye como respaldo las capturas de los cuatro dashboards aprobados.

Figura C.1: Acta de aprobación del diseño visual firmada por la administradora — Página 1

HOLDING WAKI S.A.C.

Arequipa, Perú

ACTA DE APROBACIÓN DEL DISEÑO VISUAL DE DASHBOARDS

En la ciudad de Arequipa, la administradora de Holding Waki S.A.C., identificada al pie del presente documento, declara haber revisado y aprobado el diseño visual de los dashboards implementados en Microsoft Power BI como parte de la solución de inteligencia de negocios desarrollada por el bachiller **Gonzalo Urrutía Quequezana**, correspondiente a su tesis titulada: *"Implementación de una solución de inteligencia de negocios para optimizar la generación de reportes de seguimiento de microcréditos en una empresa financiera en la ciudad de Arequipa"*.

La aprobación comprende los cuatro dashboards implementados en producción:

- **Reporte Gerencial — Empresas:** vista consolidada de todas las empresas con créditos activos.
- **Reporte Gerencial — Empresas Detalle:** análisis detallado por empresa específica mediante filtro desplegable.
- **Reporte Operativo — Cobranza:** seguimiento diario de cuotas vencidas y próximas a vencer.
- **Reporte Operativo — Alertas y Riesgo:** gestión del riesgo crediticio y cartera en mora.


Se declara que el diseño visual, la distribución de los elementos (tarjetas de KPIs, gráficos, tablas y filtros), la paleta de colores, la tipografía y la organización general de los dashboards son adecuados, claros y comprensibles para el uso diario en las funciones de seguimiento operativo y gerencial de la cartera de microcréditos de Holding Waki S.A.C.

Asimismo, se declara que los dashboards presentan la información de forma accesible desde cualquier dispositivo con conexión a internet durante el horario laboral y cumplen con los estándares esperados para la toma de decisiones en el negocio.

En señal de conformidad, se firma la presente acta.

Nota. Elaboración propia.

Figura C.2: Acta de aprobación del diseño visual firmada por la administradora — Página 2



Dayana Berro Wilena

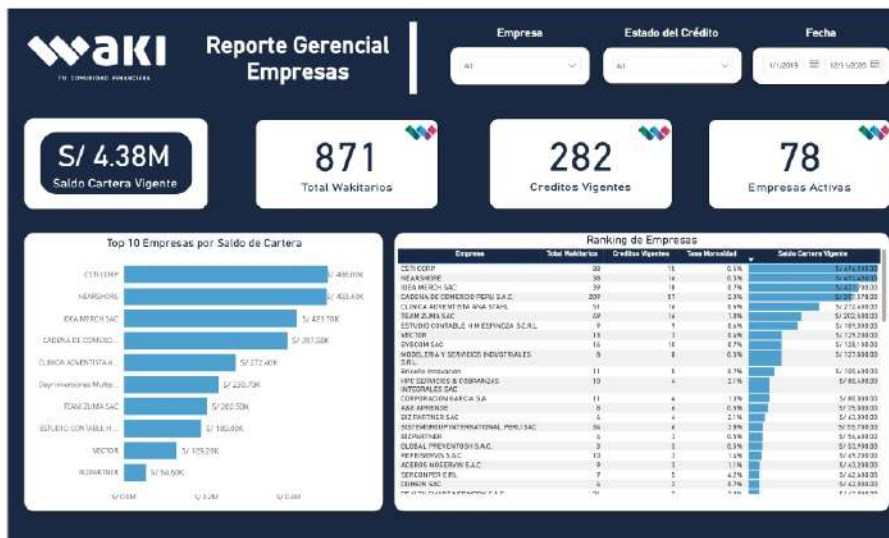
Administradora
Holding Waki S.A.C.
DNI: 47592676

Escaneado con CamScanner

Nota. Elaboración propia.

ANEXO: DASHBOARDS APROBADOS

A continuación se presentan los cuatro dashboards implementados en Microsoft Power BI cuyo diseño visual es objeto de la presente acta de aprobación.



Dashboard 1: Reporte Gerencial — Vista General de Empresas

Nota. Elaboración propia.



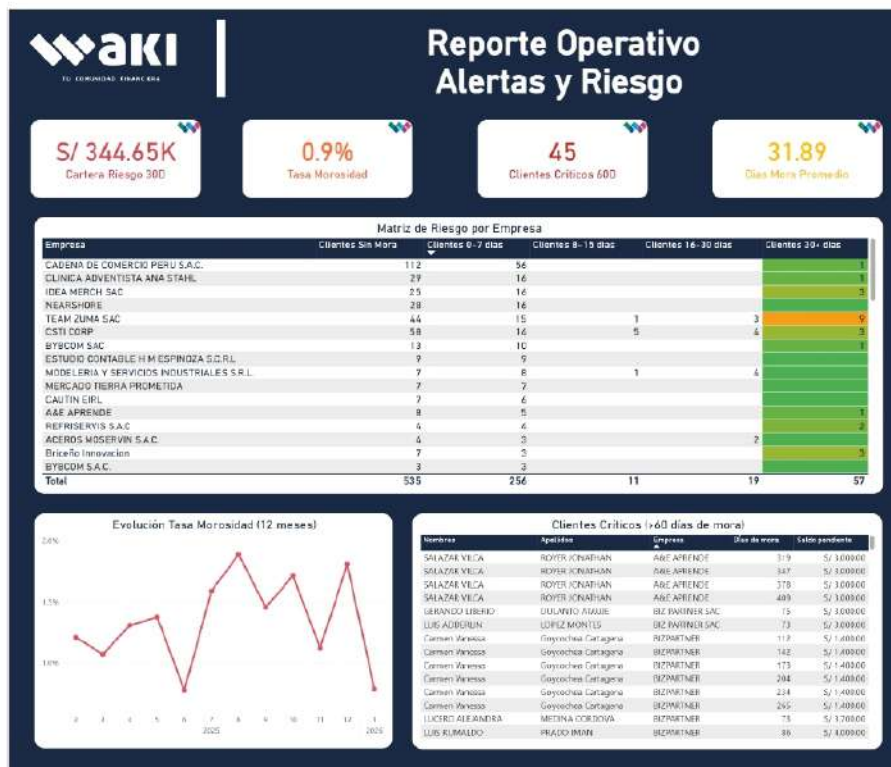
Dashboard 2: Reporte Gerencial — Detalle por Empresa

Nota. Elaboración propia.

Figura C.5: Acta de aprobación del diseño visual firmada por la administradora — Página 5



Dashboard 3: Reporte Operativo — Cobranza



Dashboard 4: Reporte Operativo — Alertas y Riesgo

Nota. Elaboración propia.

Anexo D: Pruebas unitarias de transformaciones ETL

Se implementaron 37 pruebas unitarias con *pytest* para validar la lógica de transformación del proceso ETL. Las pruebas se ejecutan sin conexión a Cassandra ni Redshift, utilizando datos de prueba contruidos localmente. A continuación se presenta el código fuente de las pruebas, organizado por componente.

D.1. Conversión de UUIDs

```
1 class TestConvertUuidsToStrings:
2
3     def test_convierte_uuids_a_strings(self):
4         """UUIDs de Cassandra se convierten a cadenas de texto."""
5         test_uuid = uuid4()
6         df = pd.DataFrame({'id': [test_uuid], 'nombre': ['Juan']})
7         result = convert_uuids_to_strings(df)
8         assert result['id'].iloc[0] == str(test_uuid)
9         assert isinstance(result['id'].iloc[0], str)
10
11    def test_dataframe_vacio_retorna_vacio(self):
12        df = pd.DataFrame()
13        result = convert_uuids_to_strings(df)
14        assert result.empty
15
16    def test_none_retorna_none(self):
17        result = convert_uuids_to_strings(None)
18        assert result is None
19
20    def test_columnas_sin_uuid_no_cambian(self):
21        df = pd.DataFrame({'nombre': ['Ana'], 'edad': [30]})
22        result = convert_uuids_to_strings(df)
23        assert result['nombre'].iloc[0] == 'Ana'
24        assert result['edad'].iloc[0] == 30
```

Listing D.1: Pruebas: `convert_uuids_to_strings`

D.2. Mapeo de tipo de producto

```
1 class TestMapProducto:
```

```

2
3 def test_waki_planilla_credito_convenio(self):
4     """Origen nulo sin bureau_msg = Crédito por Convenio (sk=1)."""
5     row = {'user_origin': None, 'user_origin_app': None,
6           'bureau_msg': None}
7     assert map_producto(row) == 1
8
9 def test_waki_planilla_adelanto_sueldo(self):
10    """Origen nulo con bureau_msg 'Adelanto sueldo' = sk=2."""
11    row = {'user_origin': None, 'user_origin_app': None,
12          'bureau_msg': 'Adelanto sueldo – Empresa ABC'}
13    assert map_producto(row) == 2
14
15 def test_waki_consumo_web(self):
16    row = {'user_origin': 'Web', 'user_origin_app': None,
17          'bureau_msg': None}
18    assert map_producto(row) == 3
19
20 def test_waki_consumo_app(self):
21    row = {'user_origin': 'App', 'user_origin_app': None,
22          'bureau_msg': None}
23    assert map_producto(row) == 4
24
25 def test_waki_consumo_simulador(self):
26    row = {'user_origin': 'Simulador Waki',
27          'user_origin_app': None, 'bureau_msg': None}
28    assert map_producto(row) == 5
29
30 def test_waki_consumo_facebook(self):
31    row = {'user_origin': 'Facebook', 'user_origin_app': None,
32          'bureau_msg': None}
33    assert map_producto(row) == 6

```

Listing D.2: Pruebas: map_producto

D.3. Mapeo de estado de pago

```

1 class TestMapEstadoPago:
2

```

```

3  def test_activo_pendiente(self):
4      assert map_estado_pago(
5          {'state': 'Activo', 'payment_state': 'Pendiente'}) == 1
6
7  def test_vencida_pendiente(self):
8      """Cuota vencida sin pago (en mora) = sk=3."""
9      assert map_estado_pago(
10         {'state': 'Vencida', 'payment_state': 'Pendiente'}) == 3
11
12     def test_finalizado_aprobado(self):
13         """Cuota pagada correctamente = sk=4."""
14         assert map_estado_pago(
15             {'state': 'Finalizado', 'payment_state': 'Aprobado'}) == 4
16
17     def test_vencida_aprobado(self):
18         """Cuota vencida con pago posterior = sk=5."""
19         assert map_estado_pago(
20             {'state': 'Vencida', 'payment_state': 'Aprobado'}) == 5
21
22     def test_pago_rechazado(self):
23         assert map_estado_pago(
24             {'state': 'Activo', 'payment_state': 'Rechazado'}) == 6
25
26     def test_refinanciado(self):
27         assert map_estado_pago(
28             {'state': 'Refinanciado', 'payment_state': 'Pendiente'}) == 7
29
30     def test_castigado(self):
31         assert map_estado_pago(
32             {'state': 'Castigado', 'payment_state': 'Pendiente'}) == 8

```

Listing D.3: Pruebas: map_estado_pago

D.4. Extracción con expresiones regulares y campos derivados

```

1  class TestExtraccionRegex:
2
3      def test_extraer_numero_cuota(self):
4          df = pd.DataFrame(

```

```

5     {'description': ['Cuota 3', 'Cuota 12', 'Pago']})
6     df['numero_cuota'] = df['description'].str.extract(
7         r'Cuota (\d+)').astype(float).fillna(0).astype(int)
8     assert df['numero_cuota'].tolist() == [3, 12, 0]
9
10    def test_extraer_dias_atraso(self):
11        df = pd.DataFrame(
12            {'delay': ['5 días', '30 días', 'Sin atraso']})
13        df['dias_atraso'] = df['delay'].astype(str).str.extract(
14            r'(\d+)').astype(float).fillna(0).astype(int)
15        assert df['dias_atraso'].tolist() == [5, 30, 0]
16
17
18    class TestCamposDerivados:
19
20        def test_monto_pagado_aprobado(self):
21            df = pd.DataFrame({
22                'payment_state': ['Aprobado', 'Pendiente'],
23                'state': ['Activo', 'Activo'],
24                'amount': [500.0, 300.0]
25            })
26            df['monto_pagado'] = np.where(
27                (df['payment_state'] == 'Aprobado') |
28                (df['state'] == 'Finalizado'),
29                df['amount'], 0)
30            assert df['monto_pagado'].tolist() == [500.0, 0]
31
32        def test_flag_vencida(self):
33            df = pd.DataFrame(
34                {'state': ['Vencida', 'Activo', 'Finalizado']})
35            df['flag_vencida'] = df['state'] == 'Vencida'
36            assert df['flag_vencida'].tolist() == [True, False, False]
37
38        def test_flag_con_mora(self):
39            df = pd.DataFrame({'payment_delay': [10.5, 0, None]})
40            df['flag_con_mora'] = df['payment_delay'].fillna(0) > 0
41            assert df['flag_con_mora'].tolist() == [True, False, False]

```

Listing D.4: Pruebas: expresiones regulares y campos derivados

D.5. Limpieza de teléfonos

```
1 class TestLimpiezaTelefonos:
2
3     def test_elimina_prefijo_51(self):
4         df = pd.DataFrame({'cell_phone': ['+51987654321']})
5         df['telefono'] = (
6             df['cell_phone'].astype(str)
7             .str.replace(r'^\+51', '', regex=True)
8             .str.replace(r'[s\ -]', '', regex=True)
9         )
10        assert df['telefono'].iloc[0] == '987654321'
11
12    def test_elimina_espacios_y_guiones(self):
13        df = pd.DataFrame({'cell_phone': ['987 654-321']})
14        df['telefono'] = (
15            df['cell_phone'].astype(str)
16            .str.replace(r'^\+51', '', regex=True)
17            .str.replace(r'[s\ -]', '', regex=True)
18        )
19        assert df['telefono'].iloc[0] == '987654321'
```

Listing D.5: Pruebas: limpieza de teléfonos en dim_cliente

D.6. Generación de dimensión tiempo

```
1 class TestGenerateDimTiempo:
2
3     def test_genera_cantidad_correcta_de_filas(self):
4         """Una semana genera exactamente 7 filas."""
5         df = generate_dim_tiempo(date(2025, 1, 1), date(2025, 1, 7))
6         assert len(df) == 7
7
8     def test_sk_tiempo_formato_yyyymmdd(self):
9         df = generate_dim_tiempo(date(2025, 3, 15), date(2025, 3, 15))
10        assert df['sk_tiempo'].iloc[0] == 20250315
11
12    def test_nombre_mes_en_espanol(self):
13        df = generate_dim_tiempo(date(2025, 1, 1), date(2025, 1, 1))
14        assert df['nombre_mes'].iloc[0] == 'Enero'
15
```

```

16 def test_trimestre_correcto(self):
17     """Enero=T1, Abril=T2, Julio=T3, Octubre=T4."""
18     fechas = [date(2025, 1, 1), date(2025, 4, 1),
19               date(2025, 7, 1), date(2025, 10, 1)]
20     for i, f in enumerate(fechas):
21         df = generate_dim_tiempo(f, f)
22         assert df['trimestre'].iloc[0] == i + 1
23
24 def test_fin_de_semana(self):
25     # 2025-01-04 es sabado, 2025-01-05 es domingo
26     df = generate_dim_tiempo(date(2025, 1, 4), date(2025, 1, 5))
27     assert df['es_fin_semana'].tolist() == [True, True]
28
29 def test_anio_bisiesto(self):
30     """2024 (bisiesto) genera 366 filas."""
31     df = generate_dim_tiempo(date(2024, 1, 1), date(2024, 12, 31))
32     assert len(df) == 366

```

Listing D.6: Pruebas: generate_dim_tiempo

Anexo E: Carta de autorización para uso de datos de la empresa

El presente anexo contiene la carta firmada por la administradora de Holding Waki S.A.C., mediante la cual se autoriza el uso del nombre de la empresa y de los datos necesarios para el desarrollo del proyecto de tesis.

Figura E.1: Carta de autorización para uso de datos de la empresa

HOLDING WAKI S.A.C.

RUC N.º 20604046590

Arequipa, mayo del 2026

CARTA DE AUTORIZACIÓN PARA USO DE DATOS DE LA EMPRESA

Asunto: Autorización para uso del nombre de la empresa y datos de usuarios en proyecto de tesis.

Yo, **Dayana Berrio Villena**, identificada con DNI N.º **47592676**, en mi calidad de administradora de la empresa **Holding Waki S.A.C.**, identificada con RUC N.º **20604046590**, por medio de la presente autorizo al tesista **Urrutia Quequezana, Gonzalo** a utilizar el nombre de la empresa **Holding Waki S.A.C.** para el presente proyecto de tesis.

Asimismo, autorizo el uso de los datos de los usuarios de la empresa que resulten necesarios para el análisis, diseño, desarrollo, validación y presentación del referido proyecto, exclusivamente con fines académicos y de investigación.

La autorización se otorga para el desarrollo, presentación, revisión y sustentación de la tesis titulada:

"Implementación de una solución de inteligencia de negocios para optimizar la generación de reportes de seguimiento de microcréditos en una empresa financiera en la ciudad de Arequipa".

El uso de dicha información deberá mantenerse dentro del alcance del trabajo académico, preservando la reserva de la información interna, sensible o confidencial que no corresponda ser difundida.

Sin otro particular, quedo de ustedes.



Berrio Villena, Dayana

DNI N.º 47592676

Administradora

Holding Waki S.A.C.

RUC N.º 20604046590

Nota. Documento proporcionado por Holding Waki S.A.C.