PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado

Diseño de una solución tecnológica para combatir el robo de terminales móviles en el Perú

Tesis para obtener el grado académico de Maestro en Ingeniería de las Telecomunicaciones que presenta:

Christian David Del Carpio Flores

Asesor:

Mag. Alan Alberto Ramírez García

Lima, 2025

Informe de Similitud

Yo, Alan Alberto Ramírez García, docente de la Escuela de Posgrado de la Pontificia Universidad Católica del Perú, asesor de la tesis titulada “Diseño de una solución tecnológica para combatir el robo de terminales móviles en el Perú”, del autor Christian David Del Carpio Flores, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 13%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 10/07/2025.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de investigación, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha:

Lima, 11 de Julio de 2025.

<table>
<tr><td colspan="2">Apellidos y nombres del asesor:
Ramírez García, Alan Alberto</td></tr>
<tr><td>DNI: 43110749</td><td rowspan="2">Firma
  </td></tr>
<tr><td>ORCID:
0000-0001-7088-249X</td></tr>
</table>

Resumen

Esta investigación aborda como problema central el “limitado nivel de recuperación de terminales móviles robados en el Perú”, evalúa la solución tecnológica actual implementada en Perú, así como propone alternativas de mejoras sustanciales de dicha solución y el fortalecimiento del modelo institucional en operación.

Para tales efectos, se realizó el análisis de las características y resultados de la solución tecnología implementada en Perú, se efectuó el estudio comparado de soluciones implementadas por otros países y se entrevistaron a expertos para validar premisas e identificar mejores prácticas internacionales.

A partir de ello, se proponen estrategias para mejorar las oportunidades y mitigar las debilidades de la solución tecnológica actual. La primera estrategia plantea la inclusión de la “Lista Gris” en el proceso de gestión de acceso a la red móvil, mediante un modelo CEIR de API sincrónico. La segunda estrategia plantea la implementación de un modelo de registro restrictivo para terminales importados y adquiridos en el extranjero que ingresan al país.

Considerando dichas estrategias, se desarrolla tanto la definición de los procesos como el diseño de los componentes de la solución tecnológica propuesta, tomando en cuenta los múltiples escenarios que el sistema afrontaría en situaciones reales de operación. Asimismo, se realizan simulaciones que permiten validar las funcionalidades del diseño propuesto.

La solución tecnológica propuesta cumple los requisitos funcionales para -potencialmente- mejorar de forma significativa la eficiencia de la solución actual. Asimismo, la investigación propone modificaciones al marco normativo vigente y busca sensibilizar a las autoridades sobre la dimensión del problema público analizado.

Abstract

This research addresses as its central problem the "limited level of recovery of stolen mobile terminals in Peru," evaluates the current technological solution implemented in Peru, and proposes alternatives for substantial improvements to said solution and strengthening of the institutional model in operation.

For such purposes, an analysis was conducted of the characteristics and results of the technological solution implemented in Peru, a comparative study was carried out of solutions implemented by other countries, and experts were interviewed to validate premises and identify international best practices.

Based on these findings, strategies are proposed to improve opportunities and mitigate weaknesses of the current technological solution. The first strategy proposes the inclusion of the "Gray List" in the mobile network access management process, through a synchronous CEIR API model. The second strategy proposes the implementation of a restrictive registration model for imported terminals and those acquired abroad that enter the country.

Considering these strategies, both the definition of processes and the design of components of the proposed technological solution are developed, taking into account the multiple scenarios that the system would face in real operational situations. Likewise, simulations are performed that allow validation of the proposed design functionalities.

The proposed technological solution meets the functional requirements to potentially significantly improve the efficiency of the current solution. Furthermore, the research proposes modifications to the current regulatory framework and seeks to raise awareness among authorities about the dimension of the public problem analyzed.

Índice

Pág

Glosario de Términos

Terminal Móvil: Dispositivo electrónico que permite efectuar comunicaciones telefónicas, de voz o datos, en una amplia zona geográfica, gracias al acceso radioeléctrico a redes móviles públicas, teniendo el usuario la posibilidad de estar en movimiento**.** Para efectos del presente documento, se referirá "terminal móvil" o "equipo móvil" de forma indistinta. Asimismo, se referirá a "sustracción" de terminal o equipo móvil, lo cual engloba situaciones asociadas al hurto o robo de dichos equipos, indistintamente.

IMEI: Identidad internacional de equipos móviles. Este identificador es único para cada equipo móvil y se encuentra asociado a únicamente a un terminal móvil.

Lista Blanca: Registro que contiene la lista de IMEI que tienen el acceso permitido a redes móviles públicas.

Lista Negra: Registro que contiene la lista de IMEI que no tienen permitido el acceso a redes móviles públicas. Cabe precisar que, a diferencia de Perú, en Colombia, dicho registro es denominado "Base de datos Negativa" o "BD –".

Lista Gris: Registro que contiene la lista de IMEI que han sido detectados en la red móvil, sin embargo, no pertenecen a la Lista Blanca ni a la Lista Negra.

Registro de identidad de equipo (EIR): Elemento en el núcleo de las redes móviles cuya función es entre otras, terminar un intento de acceso cuando se realiza un procedimiento de verificación de IMEI, según su estado, en uno de sus registros, ya sea la lista negra o lista gris**.**

Registro central de identidad de equipos (CEIR): El CEIR mantiene información sobre la elegibilidad de los dispositivos móviles para controlar el acceso a las redes móviles. El CEIR se interconecta con múltiples EIR, de modo que se mantenga un conjunto común de datos y esté disponible para los operadores participantes.

Sistema de respuesta en tiempo real: Sistema destinado a aplicaciones que cumplen requisitos de tiempo muy estrictos, con el tiempo de respuesta mínimo garantizado. En el contexto de la investigación, un EIR no espera la respuesta del CEIR para permitir o denegar una solicitud de acceso. La solicitud genera que el EIR se actualice y pueda operar autónomamente.

Glosario de Abreviaturas y Acrónimos

API: Interfaz de programación de aplicaciones

CEIR: Registro central de identidad de equipos

EIR: Registro de identidad de equipos

ESIM: Módulo de identidad del suscriptor embebida

GSMA: Asociación del sistema global para comunicaciones móviles

IMEI: Identidad internacional de equipos móviles

MNO: Operador de red móvil

JSON: Notación de objeto de JavaScript

OTP: Contraseña de un solo uso

RENTESEG: Registro nacional de equipos terminales móviles para la seguridad.

Lista de tablas

Lista de figuras

Capítulo 1 : INTRODUCCIÓN

1.1 Contexto

El robo de equipos terminales móviles es un problema público que afecta a muchos ciudadanos en todos los países latinoamericanos. Ante ello, los gobiernos, en conjunto con otros actores, han desarrollado diferentes políticas públicas y herramientas para combatir este problema.

Las referidas políticas tienen en su mayoría múltiples ámbitos de acción, siendo un caso relevante la restricción del uso de los equipos terminales móviles robados, a fin de impedir su utilización y desincentivar la compra de dichos equipos.

En el Perú, el 22 de abril de 2024, se inició la operación de un nuevo sistema que tiene como principales objetivos (i) el bloqueo inmediato -en tiempo real- de equipos terminales que sean reportados por sustracción[1] o pérdida, y (ii) la identificación y el bloqueo diario de equipos terminales alterados[2].

Estas y otras medidas generan que, en cierta proporción, la cantidad de reportes de robo de equipos terminales haya disminuido respecto a años previos, sin embargo, según las estadísticas publicadas por OSIPTEL[3], el promedio diario de reportes de robo para el año 2024 es de 4000 equipos, siendo esta cantidad alta en comparación de la cantidad de robos que se reportan en países [4] como Brasil o Colombia.

Bajo estas consideraciones, esta investigación tiene como finalidad analizar la situación actual del problema público antes descrito, considerando las herramientas y políticas públicas implementadas para disminuir la cantidad de robos de equipos terminales. Sobre esta base, se evalúan estrategias para fortalecer las oportunidades y mitigar las debilidades que se puedan encontrar.

Adicionalmente, se formula una propuesta de diseño de una herramienta tecnológica para combatir la sustracción de terminales móviles, tomando en consideración los lineamientos técnicos de la Unión Internacional de

[1] Incluye el robo o hurto de equipos terminales.
[2] Según el Decreto Legislativo 1338 y su reglamento, un IMEI alterado es aquel IMEI lógico que no tiene coincidencia con el IMEI físico
[3] Según las estadísticas publicadas por OSIPTEL en el portal PUNKU (https://punku.osiptel.gob.pe/)
[4] Según el relevamiento de información realizado para el caso de Brasil el promedio de robos diarios al 2021 es de 2382 terminales y para el caso de Colombia el promedio de robos diarios para el año 2024 es de 2037.

Telecomunicaciones y la experiencia de otros países, tales como, Colombia, Brasil e India.

Finalmente, se valida mediante la simulación, que el diseño de la solución cumple los requerimientos funcionales que permitirían combatir el robo de terminales móviles en el Perú.

1.2 Problemática

Una dimensión importante para combatir el robo de terminales móviles es la relacionada con las soluciones tecnológicas, toda vez que estas pueden ayudar a impedir que los equipos robados puedan operar en las redes de telecomunicaciones móviles de las empresas operadoras mediante diversos mecanismos factibles de implementar.

1.2.1 Definición del problema, sus causas y efectos

1.2.1.1 Causas

Causa Directa 1
Mecanismos tecnológicos ineficientes para la recuperación de los terminales móviles robados.
Causa Indirecta 1.1
Ausencia de detección en la red móvil de terminales robados para su recuperación.
Causa Indirecta 1.2
Limitada capacidad para identificar eventos efectuados en un terminal robado para su recuperación.

Causa Directa 2
Alto nivel de incentivos en el robo de los terminales móviles.
Causa Indirecta 2.1
Alta demanda de terminales móviles robados en las economías ilegales.
Causa Indirecta 2.2
Baja eficiencia de las medidas implementadas por el estado peruano para impedir la comercialización de terminales robados.

Causa Directa 3
Ineficiente diseño institucional de las organizaciones públicas competentes en la atención de este problema.

Causa Indirecta 3.1

Limitada acción por parte del Ministerio Público y Policía Nacional en la recuperación de terminales móviles.

Causa Indirecta 3.2

Ausencia de la SUNAT en el marco normativo vigente, como un actor principal en el proceso de ingreso de terminales móviles a Perú.

Causa Indirecta 3.3

Enfoque limitado en la gobernanza para una mayor tasa de recuperación de los terminales robados.

1.2.1.2 Problema central

Limitado nivel de recuperación de terminales móviles robados en el Perú.

1.2.1.3 Efectos

Efecto Directo 1

Bajo nivel de confianza en las autoridades y la acción del estado.

Efecto Directo 2

Alto nivel de pérdidas económicas y de condiciones adversas para el desarrollo económico.

1.2.1.4 Efecto final

Altas restricciones para catalizar el desarrollo socioeconómico de los ciudadanos y el ejercicio de sus derechos con base en las TIC.

1.2.1.5 Árbol de problemas

El arbol de problemas se muestra en la Figura 1:

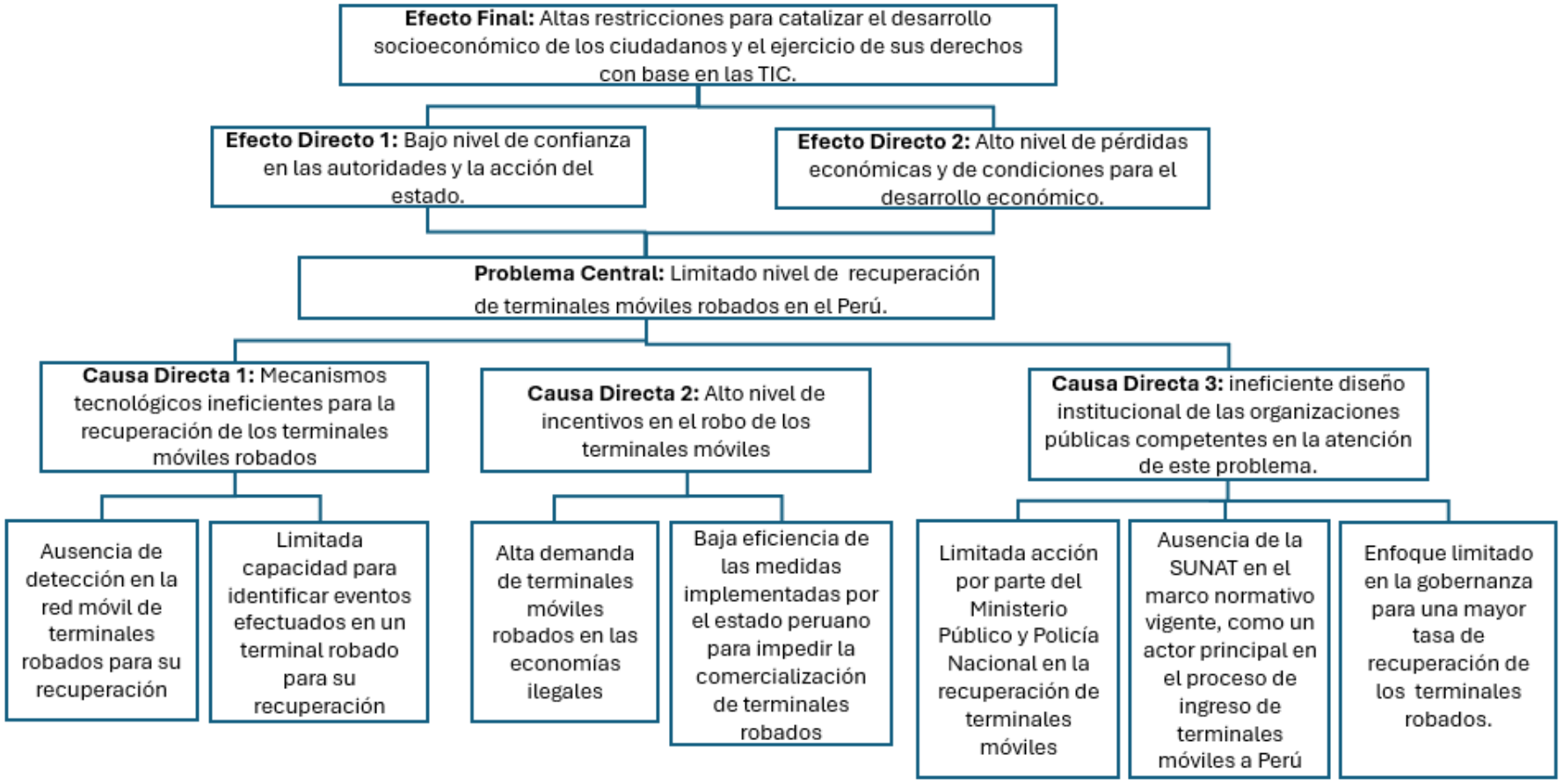


Figura 1 - Árbol de problemas
Fuente y elaboración propia

1.2.2 Definición medios y fines

Se modifica las condiciones negativas en el árbol de problemas para transformarlas en condiciones positivas en un nuevo árbol de objetivos.

1.2.2.1 Objetivo central

Se plantea el siguiente objetivo: “Alto nivel de recuperación de terminales móviles robados en el Perú.”

1.2.2.2 Medios

Así los medios para lograr el objetivo central son los siguientes:

Medio de Primer Nivel 1
Mecanismos tecnológicos eficientes para la recuperación de los terminales móviles robados. Para su realización, se presentan los siguientes medios:

Medio de segundo Nivel 1.1
Presencia de detección en la red móvil de terminales robados para su recuperación.

Medio de segundo Nivel 1.2
Amplia capacidad para identificar eventos efectuados en un terminal robado para su recuperación.

Medio de Primer Nivel 2
Bajo nivel de incentivos en el robo de los terminales móviles. Para su realización, se presentan los siguientes medios:

Medio de segundo nivel 2.1
Baja demanda de terminales móviles robados en las economías ilegales.

Medio de segundo nivel 2.2
Alta eficiencia de las medidas implementadas por el estado peruano para impedir la comercialización de terminales robados.

Medio de Primer Nivel 3
Eficiente diseño institucional de las organizaciones públicas competentes en la atención de este problema.

Medio de segundo nivel 3.1
Amplia acción por parte del Ministerio Público y Policía Nacional en la recuperación de terminales móviles.

Medio de segundo nivel 3.2
Participación de la SUNAT en el marco normativo vigente, como un actor principal en el proceso de ingreso de terminales móviles a Perú.

Medio de segundo nivel 3.3:
Enfoque extenso en la gobernanza para una mayor tasa de recuperación de los terminales robados.

1.2.2.3 Fines

En este contexto los fines identificados en el análisis de los objetivos son los siguientes:

Fin Directo 1
Alto nivel de confianza en las autoridades y la acción del estado.

Fin Directo 2
Bajo nivel de pérdidas económicas y de condiciones adversas para el desarrollo económico.

1.2.2.4 Fin del proyecto

Bajas restricciones para catalizar el desarrollo socioeconómico de los ciudadanos y el ejercicio de sus derechos con base en las TIC.

1.2.2.5 Árbol de objetivos

Se muestra el arbol de objetivos en la Figura 2:

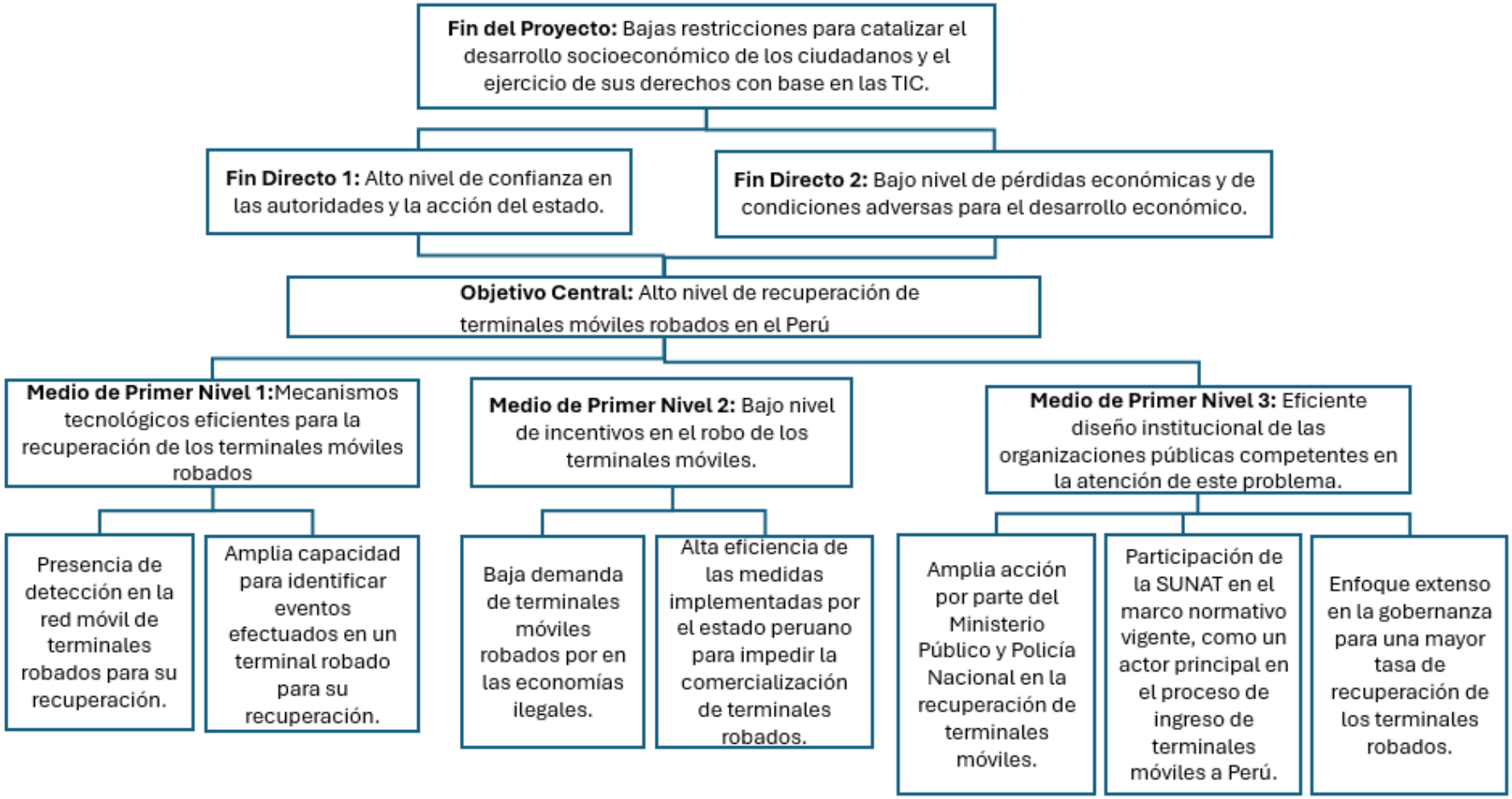


Figura 2 - Árbol de objetivos
Fuente y elaboración propia

1.2.2.6 Medios Fundamentales

Para finalizar, se identifican los medios fundamentales sobre los cuales se enfocan las alternativas de solución:

Medio fundamental 1
Presencia de detección en la red móvil de terminales robados para su recuperación.
Medio fundamental 2
Amplia capacidad para identificar eventos efectuados en un terminal robado para su recuperación.
Medio fundamental 3
Participación de la SUNAT en el marco normativo vigente, como un actor principal en el proceso de ingreso de terminales móviles a Perú.

1.3 Determinación de Objetivos

Considerando el análisis de la problemática, se plantean los objetivos del presente trabajo:

Objetivo General:

Diseñar una nueva solución tecnológica para combatir el robo de terminales móviles en el Perú.

Objetivos Específicos:

- Analizar la situación actual de la solución tecnológica para combatir el robo de terminales móviles en Perú.
- Definir los procesos para la propuesta de solución, que incluye el diseño de la gestión de acceso a la red móvil que permita la trazabilidad luego de un robo.
- Definir el modelo de las bases de datos de Lista Negra, Lista Gris y Lista Blanca para permitir el acceso a la red móvil mediante el registro centralizado de identidades de equipo de tipo sincrónico (en adelante, CEIR sincrónico).
- Definir el modelo de la base de datos para la trazabilidad de terminales móviles y su posterior recuperación.
- Diseñar la aplicación para el intercambio de información entre un EIR y el CEIR sincrónico para la trazabilidad en el acceso a la red móvil de equipos que no se encuentran en Lista Blanca o se encuentran en Lista Gris
- Realizar validaciones funcionales de la solución tecnológica propuesta, observando si el sistema implementado cumple los requerimientos funcionales que permitan combatir el robo de terminales móviles en el Perú.

Capítulo 2 : MARCO TEÓRICO

2.1 Antecedentes

En el documento denominado “IMEI-based Mobile Device Tracking and Stolen Phone Identification System” [1] Mandela et al, escriben diversas metodologías de soluciones de software que son instaladas en los propios terminales móviles. Estas soluciones son usadas cuando los usuarios sufren robos para bloquear los dispositivos o borrar la información.

La hipótesis de dicha investigación es que estas soluciones no combaten el robo ya que tienen un enfoque ex post. Considerando ello, la propuesta de los autores corresponde al diseño de un sistema que permita la trazabilidad del terminal móvil mediante la información que se genera en la red móvil.

Asimismo, el Instituto Federal de Telecomunicaciones de México realizó, mediante el documento denominado “Estudio para identificar acciones para disminuir el robo de equipos terminales móviles” [2], la identificación de acciones para disminuir el robo de equipos terminales móviles. En este estudio se analizan las diversas normativas de los países Argentina, Colombia, Chile, México y Perú respecto a la implementación de listas negras, listas blancas y listas grises como herramientas para disminuir el robo de equipos. Las conclusiones del estudio señalan que las opciones que se podrían aplicar en México son las siguientes:

- La activación obligatoria en los equipos móviles de una herramienta de bloqueo por software en todos los terminales que operen en este país. Sin embargo, se señala como una desventaja que mediante herramientas de Hacking se podría vulnerar este software y la solución no sería efectiva.
- La implementación de una lista blanca, aunque señala que generaría elevados costos de desarrollo y mantenimiento.

Finalmente, en el informe de “Simplificación del Marco Regulatorio para la restricción de equipos terminales hurtados” [3], la Comisión de Regulación de Comunicaciones de Colombia efectúa recomendaciones para la simplificación de la regulación asociada a la restricción de terminales móviles sustraídos analizando tres escenarios:

- Statu quo, es decir mantener la solución basada en lista negativa, lista positiva y los procesos de detección de IMEI Clonados, Inválidos, No Homologados y reportados por robo.
- Eliminación proceso de registro en lista positiva.

- Eliminación proceso de detección y bloqueo de IMEI no registrados en lista positiva.

2.2 Situación actual en otros países

Considerando que el problema de sustracción de terminales móviles se presenta a nivel global, se inicia la investigación revisando la experiencia de Colombia, Brasil e India, países que han implementado soluciones tecnológicas y publicaron los resultados obtenidos luego de las medidas adoptadas.

2.2.1 Experiencia comparada internacional

Existen diferentes mecanismos de identificación de terminales móviles implementados a nivel internacional; sin embargo, en el Perú y en todos los países de Latinoamérica el identificador IMEI (Identidad internacional de equipos móviles) es usado tanto para identificar un terminal móvil en la red móvil, así como para impedir su uso en caso de que un equipo es reportado como robado.

El componente de la red móvil utilizado por las empresas operadoras para permitir o denegar el acceso de los terminales móviles es el Registro de identidades de equipo, conocido como EIR. Este registro se utiliza para finalizar un intento de acceso o una llamada en curso al realizar un procedimiento de verificación de IMEI, dependiendo del estado del IMEI en uno de sus registros, sea que esté en Lista Blanca, Lista Negra o Lista Gris [4].

El proceso de verificación de IMEI se realiza cuando ocurre los siguientes eventos [5]:

- IMSI attach: Ocurre cuando se inserta o retira la tarjeta SIM
- Location Update o Actualización de la Ubicación: función de roaming.

En la Figura 3 se muestra los elementos de la red móvil que interactúan en el proceso de verificación para el acceso a la red.

Figura 3 – Flujo proceso de verificación de IMEI
Elaboración propia, fuente [6]

Asimismo, parte de las soluciones usadas en algunos países para combatir el robo de equipos terminales es el registro centralizado de identidades de equipo denominado CEIR. Este componente se interconecta con múltiples EIR para mantener un conjunto de datos común para validar el acceso de los terminales móviles en las redes móviles de las empresas operadoras participantes [6].

En la Figura 4, se muestra la interacción de los EIR de las empresas operadoras con un CEIR.

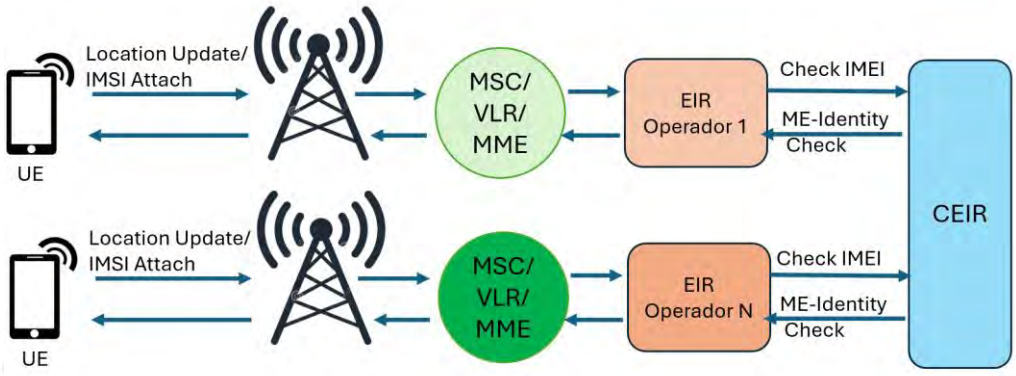


Figura 4 – Flujo proceso de verificación de IMEI en CEIR
Elaboración propia, fuente [6]

Si bien es posible que el CEIR intercambie información en tiempo real o periódicamente, el proceso de operación sincrónico (en tiempo real) permite que un reporte de robo pueda generar un bloqueo inmediato denegando el acceso a la red móvil.

2.2.1.1 Sistemas de bloqueo de actualización diaria

A continuación, se documenta la experiencia de dos países de la región considerando a Brasil debido a que es el país de con mayor cantidad de población y de participación activa en los grupos de estudio de la Unión Internacional de Telecomunicaciones para combatir el robo de terminales móviles, y a Colombia por ser miembro de la Comunidad Andina y su experiencia en la implementación de una solución tecnológica inicio en el año 2013, así durante este tiempo este país a realizado diversos estudios respecto al impacto de las medidas adoptadas.

República Federativa de Brasil

En 2004 se implementó la solución CEMI - *Cadastro de Estações Móveis Impedidas,* para gestionar el reporte de robos de terminales en Brasil.

El objetivo principal de la solución CEMI es brindar al usuario de un terminal móvil un mecanismo para bloquear su equipo en situaciones en las que el dispositivo sea robado o perdido [8].

Hasta diciembre de 2023, los usuarios de dispositivos móviles podían solicitar el bloqueo del dispositivo a las autoridades policiales o directamente a su operador móvil; sin embargo, luego de esta fecha se incluyó como una forma de bloqueo el uso de la aplicación móvil "Celular Seguro", a través del cual se realizan reportes que viajan directamente a CEMI Nacional y también son compartidos a bancos, de manera que estas entidades puedan suspender el acceso a las cuentas de las personas que habrían sido víctimas de robo.

En la Figura 5 se muestra cómo es el proceso de bloqueo de terminales móviles en Brasil incluyendo el método de bloqueo de la aplicación "Celular Seguro".

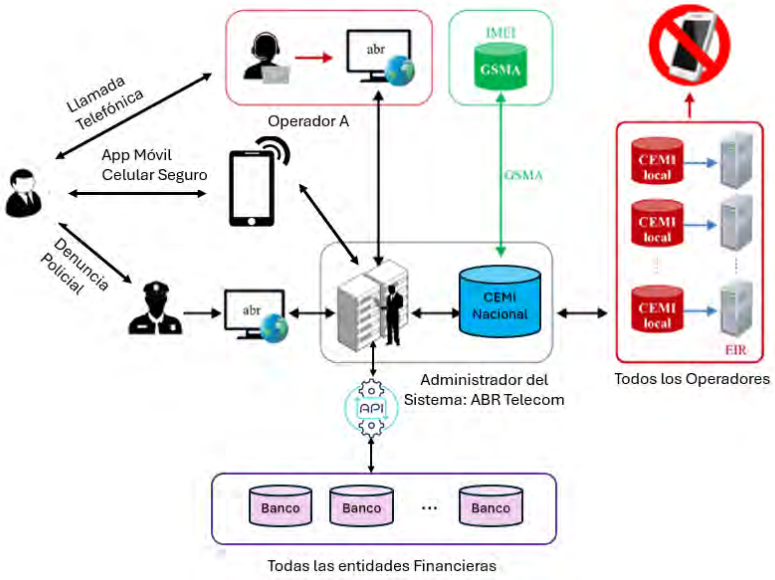


Figura 5: Proceso de bloqueo de terminales en Brasil
Elaboración propia, fuente [8]

Respecto a la cantidad de bloqueos efectuados a causa de la implementación de CEMI, en la Tabla 1 se muestra información de la cantidad de reportes de robo por año y el promedio diario de reportes de robo.

Tabla 1 – Estadística de efectos de implementación de CEMI

	2017	2018	2019	2020	2021*
Cantidad de robos de equipos por año	1473,756	1,468,836	1,393,223	1,010,886	721,818
Cantidad promedio de robos por día	4,038	4,024	3,817	2,770	2,382

Elaboración propia, fuente [8]

***** La información se muestra a octubre de 2021

Asimismo, es necesario considerar la cantidad de servicios móviles en Brasil, a fin de determinar la proporción de los robos respecto a los terminales que operarían. Considerando ello, en la Tabla 2, se muestra la cantidad de servicios móviles activos por año.

Tabla 2 – Cantidad de servicios móviles activos por año

	2018	2019	2020	2021	2022
Cantidad de Servicios Móviles	226 millones	226 millones	234 millones	254 millones	251 millones

Elaboración propia, fuente [9]

En esa línea, el Ministerio de Justicia y Seguridad Pública de Brasil informó en julio de 2024 que, a más de seis meses de su lanzamiento, el programa Celular Seguro alcanzó la cifra de 60 mil bloqueos relacionados con la pérdida, robo o hurto de terminales móviles; asimismo se menciona que por ejemplo para el estado de Piauí, en el primer trimestre de 2024, en comparación con el mismo período del año 2023, el delito de robo de celulares disminuyó un 44% en el estado. La tasa de recuperación de terminales aumentó en un 139%. [10]

República de Colombia

En Colombia, de acuerdo con la “encuesta de convivencia y seguridad ciudadana”, a un 8,7% y 8,0% de las personas de más de quince (15) años les fueron sustraídos sus terminales móviles en los años 2020 y 2021, respectivamente. Asimismo, señala que, el terminal móvil fue el objeto más sustraído a estas personas, representando un 81.7% y 81.9% en los referidos años [11].

Las políticas públicas desarrolladas se basan en una estrategia comprende varios frentes:

1. Reducir las vulnerabilidades de mercado
2. Atacar directamente las economías ilegales
3. Sensibilizar a la población sobre los daños que genera la compra de terminales sustraídos.

Considerando ello, en la Figura 6 se detalla el proceso de registro de robo de celulares en Colombia.

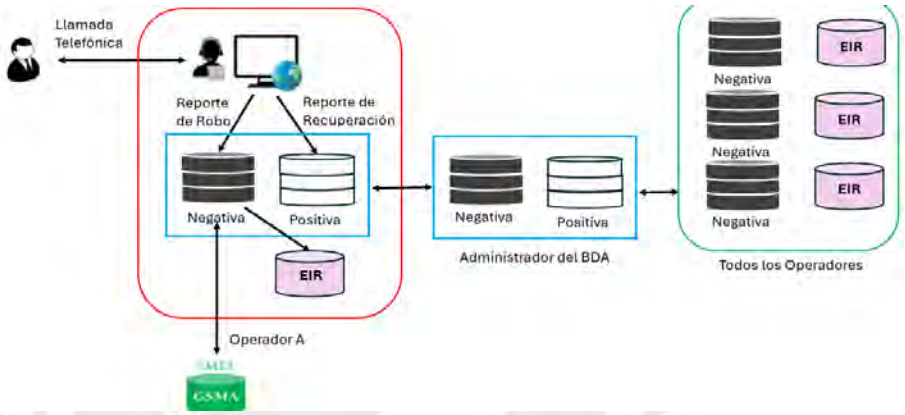


Figura 6: Proceso de bloqueo de terminales en Colombia
Elaboración propia, fuente [3]

Asimismo, en relación con la cantidad de bloqueos efectuados a causa de la implementación de la BD Positiva (Lista Blanca) y la BD Negativa (Lista Negra) en la Tabla 3 se muestra la cantidad de reportes de robo de terminales móviles por mes y por día.

Tabla 3 Estadística de efectos de implementación de BD+ y BD

	2019	2020	2021	2022	2023	2024
Cantidad promedio de robos por mes	102 922	72 202	81 558	88 145	79 076	61 782
Cantidad promedio de robos por día	3384	2374	2681	2898	2600	2037

Elaboración propia, fuente [3]

2.2.1.2 Sistemas de bloqueo sincrónico

Se considera a India, debido a que se encuentra en implementación el modelo de bloqueo sincrónico.

Experiencia en India

En India, la implementación de CEIR con el modelo de intercambio periódico comenzó en 2018 y fue completada en 2023; la implementación de intercambio en tiempo real está en progreso [7].

El procedimiento para el bloqueo publicado en la web del CEIR de la Republica de India es [12]:

- Presentar una denuncia ante la policía y conservar una copia de la denuncia.
- Obtener una tarjeta SIM duplicada para el número perdido de su proveedor de servicios de telecomunicaciones. Se debe proporcionar este como el número de teléfono móvil principal (se enviará un OTP[5] a este número) al enviar la solicitud para bloquear su IMEI.
- Cargar en la web una copia del informe policial y una prueba de identidad. También proporcionar la factura de compra del móvil.
- Registrar la información requerida por la solicitud de registro para bloquear el IMEI del terminal móvil sustraído y proceder a su envío.
- Luego de ello, se recibe el código de la solicitud, el que permite llevar una trazabilidad del trámite realizado.

En este sentido, si bien el gobierno de dicho país está desplegando esfuerzos en la implementación de un CEIR sincrónico, el procedimiento para el reporte de robo no puede ser considerado un reporte que tenga actuación en tiempo real sobre el bloqueo del equipo terminal.

Respecto a la cantidad de robos registrados en CEIR, es necesario mencionar que el acceso público del gobierno de India muestra información acumulada por cada región de India, en la Tabla 4 se muestra la cantidad de reportes de robo acumulada a Julio de 2024.

[5] Contraseña de un solo uso

Tabla 4 – Estadística efectos de implementación de CEIR

	Región	Dispositivos Bloqueados a julio 2024	Dispositivos Bloqueados a diciembre 2024	Recuperaciones a diciembre 2024	Efectividad (%)	Población estimada al 2021
1	NCT Delhi	569,402	685,081	7,125	1	32,065,760
2	Maharashtra	272,359	324,225	30,371	9	122,153,650
3	Karnataka	271,946	314,062	61,071	19	64,225,000
4	Telangana	219,734	265,760	56,239	21	35,193,978
5	Uttar Pradesh	90,006	120,647	16,501	14	237,882,725
6	West Bengal	66,399	86,757	7,348	8	99,609,303
7	Andhra Pradesh	62,855	80,098	16,565	21	53,903,000
8	Tamil Nadu	62,760	90,763	15,800	17	77,210,441
9	Punjab	51,114	90,763	3,912	4	29,362,000
10	Rajasthan	50,716	68,532	15,043	22	76,650,602
11	Chhattisgarh	43,647	58,943	12,919	22	29,362,000
12	Madhya Pradesh	43,384	61,138	10,311	17	85,358,965
13	Otras Regiones	215,519	251,618	38,874	0	
Total		**2,019,841**	**2,498,387**	**292,079**		

Elaboración propia, fuente [12]

Así también, de la información publicada en agosto de 2023 por el Departamento de Telecomunicaciones a cargo de la implementación del CEIR sincrónico, en cuidades como Telangana se recuperan el 67% de terminales móviles sustraídos, y en las cuidades como Karnataka y Andhra Pradesh se recuperan el 54 % y el 50 % respectivamente. [13]

Asimismo, en la Tabla 5, se muestra la cantidad de servicios móviles en India.

Tabla 5 – Cantidad de servicios móviles activos por año

	A junio de 2023	A junio de 2024
Cantidad de Servicios Móviles	1,146 millones	1,171 millones

Elaboración propia, fuente [14]

2.2.1.3 Registro de información de terminales móviles y gestión de acceso a la red móvil mediante la lista blanca.

La información de los equipos legalmente importados o fabricados es información de suma relevancia para la generación de políticas públicas. Así, dicha información es

importante para, por ejemplo, determinar la cantidad de ciudadanos que puedan hacer uso de servicios de gobierno digital que requieran validación facial, así como, para identificar los terminales móviles que son importados legalmente y pagan los impuestos correspondientes. Esta información también puede ser utilizada para realizar la trazabilidad de los terminales móviles que fueron alterados luego de un robo.

Considerando ello, algunos países utilizan una base de datos de registro de dispositivos que contiene el IMEI legalmente importado/adquirido junto con el documento de identificación del propietario, a manera de identificar aquellos dispositivos que han realizado un proceso legal de importación, adquisición en distribuidores locales o si los equipos fueron adquiridos en el extranjero. Otros países utilizan una base de datos de registro de dispositivos que contiene los IMEI legales importados/adquiridos sin requerir la identificación del usuario/propietario [8].

2.2.1.3.1 Enfoque del método permisivo

Experiencia de Colombia

En la actualidad debido a que aún no se aprueba la propuesta de simplificación normativa detallada en el documento “Simplificación del Marco Regulatorio para la restricción de equipos terminales hurtados” [3], en Colombia se permite que los usuarios de los terminales móviles puedan realizar el proceso de registro ante la empresa operadora que le brinda el servicio móvil. Al respecto, si bien al realizar las validaciones correspondientes para el registro se puede identificar si los equipos son alterados, no sería posible determinar si estos equipos provienen del comercio ilegal.

Considerando ello, este enfoque permisivo genera que la validación antes mencionada deba considerar mecanismos tecnológicos para evitar el fraude o la manipulación de información; sin embargo, a la fecha en Colombia las validaciones se realizan visualmente, es decir, se realiza manualmente por el personal encargado en cada empresa operadora o

por aplicaciones móviles, que no realizan una validación de la no alteración del IMEI.

2.2.1.3.2 Enfoque restrictivo

Experiencia de India

Según el enfoque restrictivo, todo celular importado o fabricado debe ser registrado al ingresar al país. Para ello este país implementó una web (https://icdr.ceir.gov.in/ivs/) para realizar el registro de terminales, que incluye el detalle de todos los terminales móviles como el modelo, marca, imei, datos del punto de ingreso.

En la información pública no se encuentra información del procedimiento que debe seguir un usuario cuando adquiere un terminal en el extranjero; sin embargo, el enfoque restrictivo en la gestión de acceso a la red no permitiría la operación de terminales no registrados en Lista Blanca.

2.3 Situación actual en el Perú

2.3.1 Marco normativo vigente para combatir el robo de terminales móviles en Perú

Por Decreto Legislativo N° 1338, se creó el Registro Nacional de Equipos Terminales Móviles para la Seguridad (RENTESEG), el cual -conforme a lo establecido en su artículo 1- está orientado a la prevención y combate del comercio ilegal de equipos terminales móviles y al fortalecimiento de la seguridad ciudadana.

2.3.1.1 Características de la lista negra

Según lo establecido en el Decreto Supremo N° 007-2019-IN -Reglamento del Decreto Legislativo N°1338- la lista negra está conformada por aquellos terminales móviles:

- Identificados como sustraídos, perdidos e inoperativos, incluyendo aquellos que son reportados en otros países con los que Perú tiene acuerdos internacionales.
- No registrados en la Lista Blanca y que estén generando tráfico en la red móvil, así como aquellos no incluidos en la Lista de Excepción[6].

[6] La Lista de Excepción es un registro en el EIR que permite que un terminal móvil pueda operar a pesar de encontrarse en lista negra.

- Con IMEI alterados, así como aquellos que se bloquean a solicitud de una autoridad competente.

Adicionalmente, se establece que, ante un reporte de sustracción o pérdida, previa consulta -en línea- al sistema del RENTESEG, se suspende el servicio y bloquea el terminal móvil reportado.

2.3.1.2 Características de la lista blanca

Según lo establecido en el Decreto Supremo N 007-2019-IN, la lista blanca está conformada por aquellos terminales móviles:

- Legalmente importados y registrados como tal en el sistema habilitado por el RENTESEG[7].
- Ensamblados o fabricados en el Perú para su venta en este país.
- Adquiridos en el exterior sin fin comercial. El abonado registra este tipo de equipos ante las empresas operadoras, para la habilitación de su uso.

2.3.2 Características de la solución tecnológica actual

Luego de revisar el instructivo técnico para la implementación del RENTESEG, aprobado por Resolución 00136-2024/GG-OSIPTEL, se observan tres procesos principales que caracterizan la solución:

2.3.2.1 Registro de información de terminales para que se permita la operación en las redes móviles de Perú.

El OSIPTEL, desde el año 2020, cuenta con un sistema WEB para el registro de equipos terminales móviles que son importados al Perú. En abril de 2024, este sistema se integró al sistema RENTESEG y los IMEI que son válidos son ingresados a la lista blanca.

Asimismo, según el instructivo técnico del RENTESEG, los importadores tienen la obligación de registrar la información de los IMEI, el país de origen, la declaración única de aduanas, entre otros.

[7] El acceso se realiza mediante la web https://registroequipolistablanca.renteseg.osiptel.gob.pe/home.xhtml

Adicionalmente, los usuarios de terminales móviles adquiridos en el extranjero sin fines comerciales, deben asistir a un centro de atención de la empresa operadora a solicitar el registro del terminal móvil a la lista blanca.

Es importante resaltar que, las validaciones que se realizan sobre los IMEI, antes del ingreso a la lista blanca, son:

a) El IMEI no debe estar incluido en la Lista Negra.
b) El IMEI debe ser válido según la base de datos de la GSMA.
c) El IMEI no debe estar alterado (coincidencia entre IMEI físico y lógico).

Luego de estas validaciones, los IMEI reportados son ingresados a la lista blanca del RENTESEG y podrán operar en las redes móviles de Perú.

Es necesario señalar que, actualmente no existe una herramienta que permita realizar consulta a los usuarios si un IMEI se encuentra registrado en lista blanca, asimismo los equipos que las empresas operadoras realizan validaciones visuales para registrar un terminal móvil adquirido en el extranjero.

2.3.2.2 Proceso de registro de reportes de robo de terminales móviles en Perú.

En junio de 2019, inició la operación del sistema RENTESEG para el intercambio de información de los equipos terminales que son reportados como sustraídos o perdidos. Las normas para la implementación de este sistema[8] establecieron que el intercambio de información para el bloqueo de celulares se realice con una frecuencia diaria.

Asimismo, el 22 de abril de 2024, el OSIPTEL informó que se dio inicio al intercambio de información en tiempo real. Como consecuencia de ello, todo reporte de sustracción o pérdida genera un bloqueo inmediato en las redes de todas las empresas operadoras y el registro del IMEI asociando en la lista negra.

[8] Normas completarías para la implementación de la Tercera Fase del RENTESEG aprobadas mediante la resolución de Consejo Directivo N° 007-2020-CD/OSIPTEL

En la Figura 7 se describe el proceso de reporte de un terminal móvil luego de un robo.

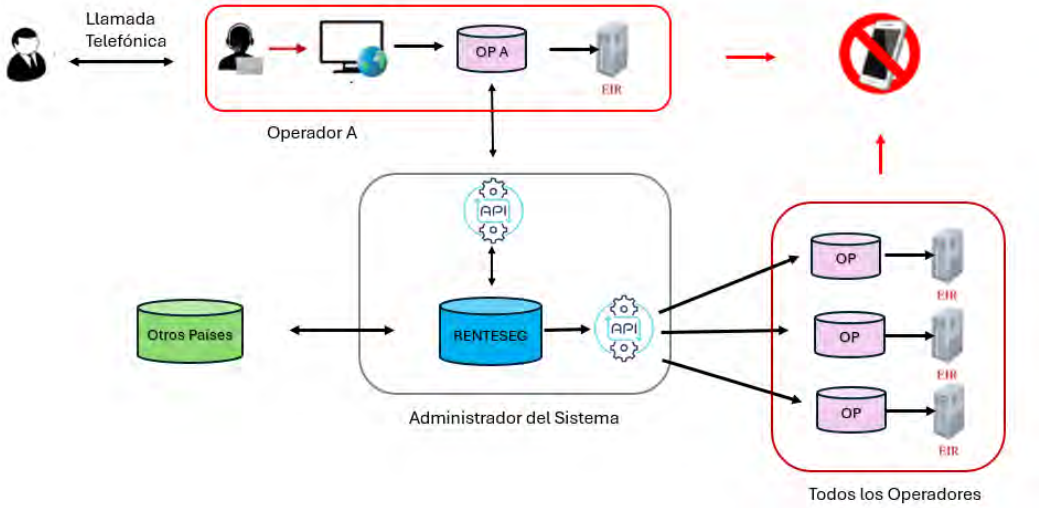


Figura 7: Proceso de bloqueo de terminales móviles robados en Perú
Elaboración propia, fuente [15]

2.3.2.3 Gestión de acceso a la red móvil mediante uso de lista negra y lista blanca.

Respecto a la gestión de acceso a la red móvil, se observa que las empresas operadoras en Perú usan sus EIR, para restringir el acceso a los IMEI que se encuentran en lista negra, y si un IMEI no se encuentra en Lista Blanca es detectado y luego de 4 días es registrado en lista negra.

Para este proceso el sistema RENTESEG recibe diariamente todas las vinculaciones de terminales móviles asociadas a servicios móviles activos de las empresas operadoras.

Con esta información se realiza el proceso de detección y se sincroniza los EIR de las empresas operadoras, a fin que todos los IMEI que se reportaron vinculados y que se encuentran en lista negra sean bloqueados.

Las acciones que ordena el sistema, según el instructivo técnico del RENTESEG en este proceso son las siguientes:

- Bloqueo por IMEI reportado como sustraído o perdido.
- Bloqueo por IMEI reportado como inoperativo.
- Bloqueo por IMEI detectado como inválido.
- Bloqueo por IMEI detectado como duplicado o clonado.

- Bloqueo por IMEI no encontrado en Lista Blanca.
- Bloqueo por IMEI excede cantidad permitida adquirida en extranjero.
- Desbloqueo por motivo “Justificado por regularización de estado del equipo terminal móvil en el sistema RENTESEG”.

2.4 Solución tecnológica para combatir el robo de terminales móviles

De acuerdo a lo revisado previamente, el actual sistema implementado en Perú, denominado RENTESEG, tiene ciertas diferencias a los modelos de CEIR descritos en el documento ITU-T Q-Sup.76 de la Unión Internacional de Telecomunicaciones, siendo la principal diferencia que el sistema RENTESEG es una base de datos centralizada de – entre otros – reportes de robos que intercambia información con los EIR de las empresas operadoras en tiempo real. Sin embargo, el RENTESEG no recibe información directa de los eventos “Location Update” o “IMSI attach”, dependiendo de la correcta operación de los EIR de cada empresa operadora.

Es necesario considerar que el recibir información de los eventos “Location Update” o “IMSI attach” permitiría que el sistema pueda registrar información para ejecutar procesos de trazabilidad para la recuperación de terminales móviles robados.

Al revisar el modelo de operación de un CEIR sincrónico para la gestión de acceso a la red móvil, se observan dos componentes para la operación de esta solución que serían los siguientes:

- Base de datos centralizada
- APIs para el intercambio de información en tiempo real.

Es necesario mencionar que, debido a la cantidad de datos que recibiría la solución tecnológica y las diversas fuentes, esta se enmarcaría en una solución de big data, la cual tiene por facultad gestionar el procesamiento de un gran volumen de información, generando data válida para el desarrollo de conocimiento [16].

Para su despliegue, se plantea el uso de soluciones basadas en computación en la nube, brindando flexibilidad y menor costo de implementación y operación, además de acceso a capacidades de

cómputo y almacenamiento bajo demanda. [16]. Considerando ello, los modelos para la implementación en nube son descritos a continuación:

- Infraestructura como servicio (IaaS): Brinda a los usuarios, bajo demanda, infraestructura en la nube, siendo responsables del sistema operativo empleado y las aplicaciones implementadas.
- La plataforma como servicio (PaaS): Ofrece y administra el hardware y software para la implementación de aplicaciones en la nube. Esta plataforma puede ser gestionada sin contar con una infraestructura independiente.
- El software como servicio (SaaS): Son gestionados por el proveedor del servicio, ofreciendo aplicaciones -en la nube- para ser usadas directamente por los clientes.
- Funciones como servicio [17] (FaaS): Es una nueva forma de crear e implementar software del lado del servidor, orientada a la implementación de funciones u operaciones individuales.

2.4.1 Análisis de Big Data en tiempo real

La solución tecnológica debe tomar decisiones para brindar el acceso a red en tiempo real, para ello es necesario precisar que un sistema destinado a aplicaciones en tiempo real debe cumplir requisitos de tiempo muy estrictos, con un tiempo de respuesta mínimo garantizado [18]. Considerando ello, la gestión, el procesamiento y el análisis de la información para un adecuado funcionamiento del sistema debe cumplir las siguientes características:

2.4.1.1 Gestión y procesamiento de datos en tiempo real.

Los eventos de gestión de acceso a la red requieren respuestas en tiempo real para una operación efectiva del CEIR sincrónico, es decir el tiempo de respuesta mínimo garantizado deberá ser menor igual a 30 segundos[9], así mismo la carga masiva de terminales importados requiere un procesamiento por lotes.

Considerando ello, la arquitectura de la solución deberá unificar el procesamiento por lotes y el procesamiento en tiempo real en una plataforma común, para ello se considera las siguientes opciones:

[9] El instructivo técnico de la solución actual considera como tiempo máximo de respuesta 30 segundos.

- **Solución de mediación de eventos** [16], se encargan de gestionar la transmisión de estos entre aplicaciones, proporcionando, tanto mecanismos de publicación y subscripción, como de mensajería directa.

  Un sistema de referencia es Apache Kafka. El funcionamiento de esta solución se sustenta en la segregación de eventos en tópicos que son a su vez separados en particiones y réplicas. Esta plataforma está orientada a la distribución de eventos.

  Otras soluciones como Apache Flink, Storn o Samza están centradas en la transformación, generando nuevos flujos a partir de aquellos que entran en el sistema.

- **Solución para unificar el procesamiento** [16] las arquitecturas Lambda y Kappa se usa para unificar el procesamiento por lotes y en tiempo real, siendo Lambda, cuando domina la información histórica y, Kappa, cuando se basa principalmente en la información en tiempo real.

  Para entender esta solución, se describe el caso de un proceso de compra en una tienda web. Por ejemplo, tradicionalmente la información que registra el sistema son los pedidos y pagos realizados; sin embargo, existe también eventos que se generan en el tiempo real en las interacciones que realizan los usuarios con la tienda web (la acción de seleccionar o visualizar un artículo). Si bien, los eventos no se guardan en una base de datos transaccional, pueden procesarse para obtener información, para, por ejemplo, analizar el comportamiento de los usuarios en la web.

2.4.1.2 Análisis de datos en tiempo real

El analizar los datos a medida que son generados es uno de los principales objetivos del procesamiento en tiempo real, esto es, realizar consultas, efectuar predicciones, detectar patrones, tendencias, entre otros. Este análisis recibe el nombre de procesamiento de eventos complejos [16].

Respecto a las consultas sobre los datos, el acceso en tiempo real viene a ser la culminación de la evolución de las

capacidades de las tecnologías de análisis, llegando finalmente a un escenario donde el dato se procesa de forma continua, tomándose decisiones en base a inferencia en tiempo real.

Los requerimientos de velocidad que impone el procesamiento en tiempo real son incompatibles con la mecánica de procesar, guardar y leer el dato para consumirlo.

2.4.2 Base de datos

En este punto es necesario mencionar a los sistemas de administración de bases de datos, los cuales son herramientas basadas en software que permiten controlar el acceso, almacenar, organizar, gestionar, recuperar y mantener una base de datos. Su principal función es realizar consultas constantes hacia una base de datos donde se encuentra la información recolectada.

Asimismo, el modelo de base de datos para la solución debe considerar las siguientes características:

2.4.2.1 Datos en cuanto al formato, datos estructurados

Se considera que la fuente de información está estructurada cuando tiene esquema organizativo, tanto en lo relacionado a tipos como a significado. Un ejemplo de lo antes señalado corresponde a una base de datos SQL, donde los datos se organizan en tablas y relaciones con referencias.

2.4.2.2 Datos en cuanto a su rol

Respecto a esta característica, es necesario comprender bien el papel y su valor para el negocio ya que algunos datos son de acceso controlado por los usuarios, en algunos casos, en transacciones comerciales o para cumplir exigencias regulatorias o judiciales. Estos datos se clasifican, en cuanto a su rol, en cuatro categorías [16]:

Datos Maestros: Detallan entidades principales del negocio y son críticos para el funcionamiento. Para la solución tecnológica, se pueden considerar la información de los concesionarios móviles que intercambian información con el sistema.

Datos Operacionales: Corresponden a los derivados del propio funcionamiento. Son generados por sistemas de relevancia crítica, cuya caída implica la paralización de las actividades.

Datos Externos: Son datos vinculados al negocio, aunque no producidos por éste. Para la solución tecnológica se puede considerar datos externos, los terminales robados en otros países.

Datos Analíticos: Tiene una perspectiva dimensional. Se producen a raíz de los datos operacionales, dentro del contexto de los datos maestros.

La gestión de datos corporativos refiere al proceso de gestión de los cuatro tipos de datos antes referidos. La Figura 8 describe la interacción de dichas tipologías.

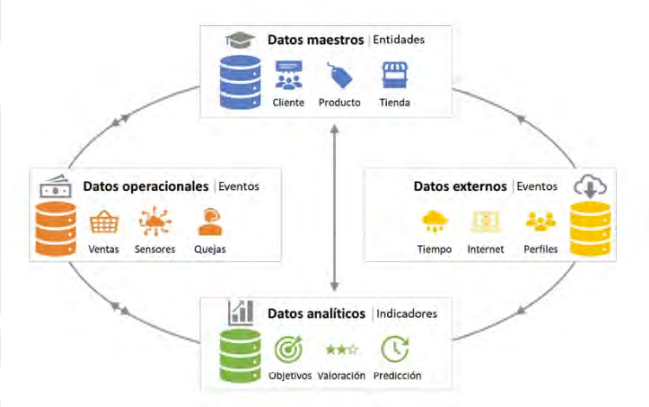


Figura 8: Proceso de gestión de datos corporativos
Fuente [16]

2.4.3 Sistemas de intercambio de información en tiempo real

De los modelos de CEIR descritos en el documento ITU-T Q-Sup.76[7], el modelo sincrónico tiene como componente fundamental para el intercambio de información al API REST que es una interfaz de programación de aplicaciones (API), alineado a los principios de diseño para sistemas hiper distribuidos del estilo arquitectónico de transferencia de estado representacional (REST) [19].

2.4.3.1 Solicitudes de API

El concepto fundamental de un sistema basado en REST es el recurso y puede ser identificado mediante un identificador

uniforme de recursos (URI). Asimismo, las operaciones que se pueden realizar con el URI junto con el método HTTP apropiado son los siguientes [20]:

- GET: usado para acceder a la colección o a un único recurso
- POST: usado en la creación de un nuevo recurso
- PUT: usado en la actualización de un recurso existente
- DELETE: usado en la eliminación de una colección o un único recurso

2.4.3.2 Cuerpo del mensaje para las operaciones del API

Un método común para que las API reciban datos de terceros es aceptar un documento JSON como cuerpo. Suponiendo que su API proporciona datos a terceros en forma de JSON, esos mismos terceros también deberían poder producir documentos JSON [20].

El documento JSON puede especificar el tipo de datos (por ejemplo, números enteros, cadenas, valores booleanos) y, al mismo tiempo, permitir relaciones jerárquicas de datos.

2.4.3.3 Autenticación y autorización

La autenticación es el proceso de verificar quién es un usuario, proporcionando, por ejemplo, un nombre de usuario y una contraseña. La autorización consiste en determinar a qué recursos tiene acceso el usuario y qué acciones puede realizar. En conjunto, se puede hacer referencia a estos dos términos como autenticación. Así existen dos paradigmas comunes en los que su API puede autenticar a los consumidores.[20].

-Paradigma de two-legged, hay dos partes involucradas: un consumidor y su proveedor. La clásica implementación es el método Basic HTTP Auth.

-En el paradigma de three-legged, hay tres partes involucradas, un consumidor, su proveedor y un usuario que tiene (o tendrá) una cuenta con ambos servicios. Así, en lugar de pasar mensajes entre dos partes (un canal), los mensajes deben comunicarse entre tres partes (tres canales) [20].

2.4.3.4 Respuestas del API

Existen muchas formas de transmitir información a los consumidores del servicio. Si se eligen métodos que sigan patrones bien entendidos, la integración con los diferentes actores será más simple.

Como regla general, la estructura de un recurso de respuesta debe parecerse mucho al recurso de solicitud equivalente. Esto significa que se utilizan los mismos nombres y valores de atributos para las solicitudes, así como para las respuestas [20].

2.5 Medición de la eficiencia del diseño de la solución tecnológica para combatir el robo de terminales móviles.

Es necesario considerar que la solución tecnología presta el servicio de mantener actualizada la información para evaluar si un determinado terminal debe operar en la red móvil de las empresas operadoras. En ese sentido el concepto de eficacia global de los equipos (OEE) se puede utilizar para medir la eficacia y el rendimiento de los procesos; asimismo proporciona información entre otros aspectos de la eficacia en la prestación de servicios [21].

El cálculo de la OEE considera tres factores en marco de la prestación de servicios:

- **Disponibilidad**: la puntuación de disponibilidad mide el tiempo de prestación del servicio.
- **Rendimiento**: la puntuación de rendimiento evalúa el rendimiento del servicio con su máximo potencial.
- **Calidad**: evalúa la tasa de respuesta sin defectos ni repeticiones.

El OEE se calcula multiplicando la disponibilidad, el rendimiento y los factores de calidad.

Así, mejorar OEE permite identificar y abordar las ineficiencias en el proceso. Al optimizar la disponibilidad, el rendimiento o la calidad, generando que las organizaciones pueden aprovechar mejor sus recursos, lo que se traduce en una mayor eficiencia general.

En ese sentido, si bien es posible conseguir una mayor eficiencia, mejorando la eficacia del proceso actual, la propuesta de esta investigación es modificar el enfoque que se tiene en el sistema actual para conseguir un mayor índice de recuperación de terminales móviles robados; asimismo, optimizar el proceso de registro de terminales.

Capítulo 3 : DISEÑO Y RESULTADOS

3.1 Análisis de la situación actual

Para la evaluación de la situación actual, se utilizaron dos herramientas:

- Entrevistas a expertos [22] y [23], como metodología cualitativa a fin de observar cómo se afronta la problemática de robo de equipos en diferentes países, las mejoras que se realizan en los procesos y la forma de terminar la efectividad de las soluciones implementadas.

- Evaluación interna y externa para determinar los procesos que se podrían mejorar, a fin incrementar la eficiencia del sistema para combatir el robo de terminales móviles en Perú, tomando como referencia las recomendaciones de la ITU-T Q-Sup.76 y la información relevada en las entrevistas a expertos.

3.1.1 Identificación de Involucrados

La problemática del robo de terminales móviles es multidimensional; sin embargo, respecto al enfoque de la propuesta de solución tecnológica es necesaria la participación de los siguientes involucrados:

-Ministerio del Interior (MINITER): En marco de sus funciones de seguridad pública, tendrán acceso a la información generada por la solución tecnología y deberá definir los procesos correspondientes a fin de ubicar a los responsables de los robos de terminales móviles.

-Ministerio de Justicia y Derechos Humanos (MINJUS): Acompañar la generación de los procesos que defina el Ministerio del Interior, respecto al proceso de recuperación de equipos.

-Ministerio Público (MP): Definir el procedimiento para actuar en los casos que en que las personas no cumplan con el proceso de recuperación establecido por el Ministerio del interior y Ministerio de justicia.

-Organismo Supervisor de la Inversión Privada en Telecomunicaciones (OSIPTEL): En su rol de administrador del sistema RENTENSEG, realizar las adecuaciones para la operatividad del proceso de recuperación de terminales móviles.

- Superintendencia Nacional de Aduanas y de Administración Tributaria (SUNAT): En marco de sus funciones, hacer cumplir el procedimiento de registro de terminales móviles adquiridos en el

extranjero y mejorar el control en el ingreso de terminales importados.

3.1.2 Análisis de fortalezas y debilidades de la solución tecnológica actual.

Considerando la metodología de análisis de factores externos, en la Tabla 6 se detallan los factores determinantes de éxito de la solución actualmente utilizada en Perú para combatir el robo de terminales móviles.

Tabla 6 – Matriz de Factores Externos

Nro	Factores determinantes de éxito	Peso	Valor	Ponderación
	Oportunidades			
1	Trazabilidad de equipos terminales legales, ya que, existen normas que obligan a las empresas importadoras de terminales móviles a realizar el registro de terminales móviles, cuando estos ingresan al país.	0.18	3	0.54
2	Las personas están de acuerdo en que se restringa el acceso a las redes móviles de equipos no registrados en la lista blanca, para combatir el robo de terminales móviles.	0.15	3	0.45
3	Se han implementado recomendaciones de UTI para enfrentar el robo de terminales móviles.	0.1	2	0.2
4	Control en el ingreso al Perú de terminales móviles adquiridos en el extranjero por personas naturales	0.12	1	0.12
5	Existe una baja tasa de recuperación de terminales móviles robados por parte de las autoridades como policía nacional.	0.12	1	0.12
6	Los terminales móviles con ESIM[10] son factibles de identificar, aun cuando se altere el IMEI	0.1	1	0.1
		0.77		**1.53**
	Amenazas			
7	Existencia de un mercado informal de venta de terminales móviles en el Perú.	0.15	4	0.6
8	La sociedad aún compra terminales móviles en lugares conocidos por vender celulares robados.	0.12	2	0.24
9	Los códigos IMEI pueden ser alterados usando ilegalmente software	0.06	2	0.16
		0.33	2	0.96
	Total	**1**		**2.49**

[10] Abreviatura de SIM embebido

Considerando lo analizado en la matriz de la tabla 6, se deben establecer estrategias para mejorar la respuesta de las oportunidades 4 y 5, siendo estas estrategias prioritarias, asimismo se respecto a las oportunidades 3 y 6 se deben también establecer estrategias para mejorar la respuesta.

De la misma manera, se realiza un análisis de factores internos del proceso operativo de la solución actual para combatir el robo de terminales móviles, el detalle del análisis se muestra en la Tabla 7.

Tabla 7 – Matriz de factores internos

Factores determinantes de éxito	**Peso**	**Valor**	**Ponderación**
Fortalezas			
Bloqueo de terminales móviles en tiempo real, luego de un reporte de robo.	0.14	4	0.56
Sistema estable y de alta disponibilidad, para recibir los reportes de robo en tiempo real.	0.12	4	0.48
Campañas de difusión para que los usuarios que sufren robos realicen los reportes y se bloquen los equipos	0.1	3	0.3
Se publica en web, para validar antes de la compra, información si un terminal móvil fue robado.	0.08	3	0.24
	0.44		**1.58**
Debilidades			
No se registra información que permita generar un proceso que ayude a la recuperación de terminales móviles robados.	0.14	1	0.14
La no habilitación de la lista blanca[11] genera que terminales móviles alterados puedan operar en las redes móviles.	0.12	1	0.12
Falta de integración de los sistemas de reporte de robos con entidades bancarias, genera que las personas que reportan un robo tengan que llamar a todos los bancos para realizar los reportes.	0.12	1	0.12
No existe una web o herramienta que permita validar si un terminal móvil se encuentra en lista blanca	0.1	1	0.1
Falta de información en el sistema para una adecuada trazabilidad para procesos de recuperación.	0.08	2	0.16
	0.56		**0.64**
Total	**1**		**2.22**

[11] El 15 de Julio de 2024, el MININTER y otros organismos acordaron suspender temporalmente el bloqueo de los equipos terminales móviles no registrados en la 'Lista Blanca' del Renteseg, hasta la aprobación de la nueva propuesta normativa, que asegurará el óptimo funcionamiento del sistema.

La matriz cuenta con 9 factores determinantes de éxito, 4 fortalezas y 5 debilidades. El valor de 2.22 indica que la solución actual es ligeramente débil. Por tanto, se deben desarrollar estrategias internas para mejorar las debilidades que tienen un peso importante.

3.1.3 Análisis de eficiencia de la solución actual en minimizar la cantidad de robos de terminales móviles.

Respecto a la cantidad de bloqueos efectuados a causa de la implementación del sistema RENTESEG, en la Tabla 8 se muestra la cantidad de reportes de robos por año y el promedio de robo por día.

Tabla 8 – Estadística de efectos de implementación de RENTESEG

	2019	2020	2021	2022	2023	2024*
Cant. de robos de equipos por año	2,191,178	1,080,115	1,351,644	1,709,770	1,707,014	668,137
Cantidad promedio de robos por día	6,003	2,959	3,703	4,684	4,677	4,454

Elaboración propia, fuente [24]

* La información se muestra a abril de 2024

Así, para determinar la eficiencia del sistema RENTESEG para disminuir la cantidad de robos de celulares en el Perú se observa el mes de mayo de 2024, y el mismo mes del 2023, en ese sentido se obtiene una eficiencia del 13%, tal como se muestra en la Tabla 9.

Tabla 9 – Calculo de la eficiencia del sistema en base a la cantidad de reportes de robo.

	Mayo 2019 (*)	Mayo 2022 (*)	Mayo 2023	Mayo 2024
Cantidad de reportes de robo por mes	208,253	151,510	150,553	131,401
Eficiencia anual en la disminución de la cantidad de robos de celulares	-	27%	1%	13%

Elaboración propia, fuente [24]

(*) No se consideran los años 2020 y 2021 debido a la cuarentena obligatoria realizada en Perú como consecuencia de la pandemia

En ese contexto, en base a la información relevada en el análisis cualitativo realizado en la entrevista a experto consultado [23], la medición de eficiencia debe considerar que es posible advertir incrementos en la cantidad de robos en el periodo de tiempo analizado, considerando que existe también incremento en la cantidad de servicios móviles activos.

3.1.4 Análisis de eficiencia de la solución actual en incrementar la cantidad de recuperaciones de terminales móviles robados

Según la información relevada en el análisis cualitativo en las entrevistas a expertos, actualmente no es posible determinar la eficiencia de la solución tecnológica implementada en Brasil [22]; sin embargo, se ha realizado un piloto, agregando a su solución el manejo de la lista gris y el uso de un procedimiento para la recuperación. El experto señala que con esta modificación sería posible determinar la eficiencia de la solución tecnológica.

En este escenario y considerando el análisis cualitativo luego de las entrevistas a expertos, se propone la siguiente fórmula para el cálculo cuantitativo de la eficiencia:

$$\text{Eficiencia} = \frac{\text{Cantidad de Recuperaciones}}{\text{Cantidad de Reportes de Robo}}$$

Al respecto, es necesario mencionar que, la solución tecnológica implementada en Perú registra información de las recuperaciones de equipos celulares que fueron bloqueados previamente por sustracción o pérdida; sin embargo, no existe un enfoque multisectorial que considere procesos que permitan incrementar las recuperaciones. Sin embargo, a fin de tener una referencia del comportamiento a nivel de cantidades se muestra la información en la Tabla 10.

Tabla 10 – Cálculo de la eficiencia del sistema en base a la cantidad de recuperaciones.

	May-19	**Mayo 2022 (*)**	**May-23**	**May-24**
Cantidad de reportes de robo por mes	208,253	151,510	150,553	131,401
Cantidad de reportes de recuperación por mes	9,645	8,179	10,216	9,695
Eficiencia en cantidad de recuperaciones de equipos robados	5 %	5 %	7 %	7 %

Elaboración propia, fuente [24]

(*) No se consideran los años 2020 y 2021 debido a la cuarentena obligatoria realizada en Perú

Se puede observar que, la eficiencia en la cantidad de recuperaciones es baja ya que como se muestra en la Tabla 10, la

eficiencia del sistema implementado en Perú para el mes de mayo 2024 es del 7%, en comparación con la eficiencia en las recuperaciones en algunas regiones de India que serían mayores al 50%, según se señala en el numeral 2.2.1.2 del presente trabajo.

3.2 Análisis y diseño

3.2.1 Diseño de los procesos

La propuesta de diseño de los procesos se realiza en base al análisis de la situación actual, considerando los aspectos más relevantes para que la solución tecnológica propuesta cumpla con el objetivo de combatir el robo de terminales móviles. A continuación, se desarrollan cada uno de los procesos:

3.2.1.1 Diseño del proceso para el registro de información desde el ingreso de un terminal móvil a Perú.

De acuerdo con lo analizado en la matriz de evaluación de factores externos, respecto a la oportunidad “Control en el ingreso al Perú de terminales móviles adquiridos en el extranjero por personas naturales” se plantea la siguiente estrategia, para mejorar la respuesta y aprovechar mejor esta oportunidad, abordando la estrategia bajo dos escenarios:

i) Equipos adquiridos en el extranjero para uso personal.

Actualmente el registro de terminales adquiridos en el extranjero se realiza registrando una declaración jurada, presentándose dicho documento en la empresa operadora que le brinda el servicio. Este procedimiento genera los siguientes problemas:

- El procedimiento de registro considera validaciones visuales del terminal móvil; sin embargo, no existe certeza del origen del equipo terminal y si dicho equipo fue alterado.
- Al ser una declaración jurada, bastaría que la persona indique que el equipo fue adquirido en el extranjero no existiendo trazabilidad del origen del equipo.

Para abordar estos problemas se plantea el siguiente procedimiento:

a) La SUNAT debe ser la entidad responsable del control de ingreso de terminales móviles adquiridos en el extranjero

para uso personal y debe trabajar junto con la Superintendencia Nacional de Migraciones (Migraciones).

b) Considerando ello, en todos los puntos de control migratorio, luego de que la persona recibe la autorización para el ingreso a Perú, deberá informar a Migraciones que ingresará un terminal móvil para uso personal, indicando un número de celular con el cual usará dicho terminal.

c) Una solución tecnológica remitirá un SMS al número celular brindado. El SMS incluirá un código OTP de duración de 5 días, además de información de la aplicación móvil que deberá instalar el usuario para realizar el registro del terminal adquirido en el extranjero.

d) Para el ingreso a la aplicación móvil, se digitará el número de servicio móvil informado al inicio del proceso y el código OTP recibido. Luego esta aplicación accederá a la siguiente información del terminal móvil: IMEI, número de serie, eUICC Identity, marca y modelo del equipo terminal y solicitará la confirmación del usuario para realizar el registro. La aplicación móvil no permitirá el acceso cuando detecte que el sistema operativo del terminal se encuentre alterado.

e) Remitida la información el sistema validará que el IMEI tenga asociación con el modelo y marca, según la base de datos de GSMA. Finalizado este proceso, el usuario recibirá un SMS de confirmación del registro o caso contrario se informará el motivo de no registro y la aplicación móvil se desinstalará luego de 1 semana.

f) Para el caso de ciudadanos extranjeros que requieran utilizar un servicio móvil de Perú, deberán presentar una declaración jurada ante la empresa operadora. Luego de las validaciones correspondientes, la empresa operadora reportará la información al RENTESEG para el registro del terminal en Lista Blanca. Este registro solo será válido como máximo por 3 meses.

En la Figura 9, se muestra la propuesta del flujo del proceso de registro de terminales móviles adquiridos en el extranjero para uso personal.

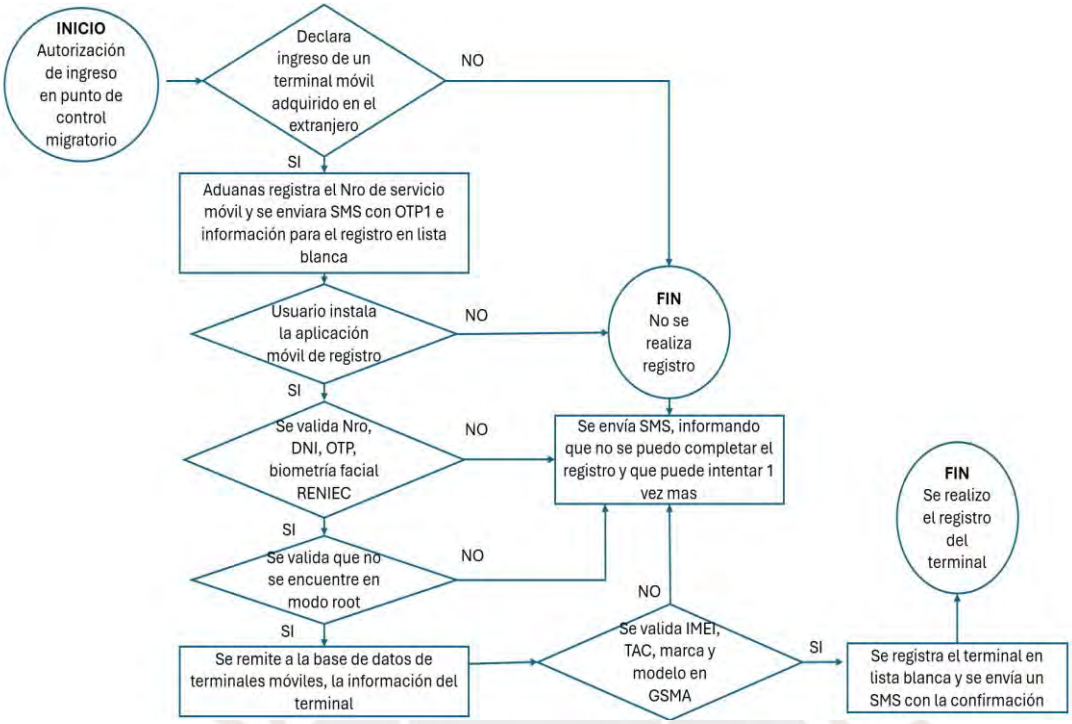


Figura 9: Propuesta de proceso de registro de terminales adquiridos en el extranjero
Fuente y elaboración propia

ii) Registro de terminales móviles importados a Perú

Luego de analizar las declaraciones realizadas por empresas importadoras respecto a los terminales móviles importados, se observa que actualmente las declaraciones que realizan las empresas importadoras ante SUNAT no registran información que permita identificar independientemente cada terminal, toda vez que, la declaración señala únicamente la cantidad de equipos, la marca y modelo.

Considerando ello, no se cuenta con información suficiente para los casos en que se requiera identificar si un importador ingreso al Perú un determinado terminal móvil.

En ese sentido, se propone que el RENTESEG mediante el sistema de registro para terminales móviles importados, incluya el registro de otro identificador único como:

- Número de Serie del terminal, el cual es asignado por el fabricante y utilizado para procesos de garantía, entre otros.

En este mismo contexto, se plantea el siguiente proceso para el registro de terminales móviles importados a Perú:

- Cuando se realiza la DAM - Declaración Aduanera de Mercancías, los importadores deben declarar la siguiente información, referida a terminales móviles:
    - Marca
    - Modelo del terminal móvil
    - Código del certificado de homologación
    - Cantidad de terminales
    - País de origen
    - RUC de la empresa importadora
    - Número de serie de cada terminal móvil

- En el desaduanaje, la SUNAT valida los números de serie declarados en la DAM para todos los terminales móviles.
- Luego, los importadores deben reportar la información de los terminales móviles en el "Sistema de Registro de Terminales Móviles Importados a Perú". El proceso de registro se realiza mediante una interfaz web, ingresando a esta con el RUC y una contraseña, considerando la siguiente información:
    - Tipo (Placa o Terminal Móvil)
    - Código DAM - Declaración Aduanera de Mercancías
    - IMEI 1
    - IMEI 2 (De corresponder, según TAC GSMA)
    - Número de serie del terminal móvil
    - Marca
    - Modelo

- El sistema realiza las siguientes validaciones:
    - IMEI1 e IMEI2 son válidos según GSMA
    - Se valida si IMEI no se encuentra en lista negra o lista gris
    - Se valida si IMEI no se encuentra en lista blanca
    - Existe correspondencia entre la marca y modelo con el IMEI1
    - Existe correspondencia entre la marca y modelo con el IMEI2
    - Se valida en línea con SUNAT si el código DAM tiene asociado el número de serie del terminal.

- En caso de que no se cumpla una de las validaciones, el registro se rechaza y debe ser subsanado lo que implica

que los terminales móviles no se registren y no deben ser comercializados.

- Si se cumplen todas las validaciones, el sistema registra el terminal en la tabla terminales importados, en lista blanca y genera un código de carga. Adicionalmente informa la cantidad de registros procesados y los registros con error.

En la Figura 10, se muestra la propuesta del flujo del proceso de registro de terminales móviles importados:

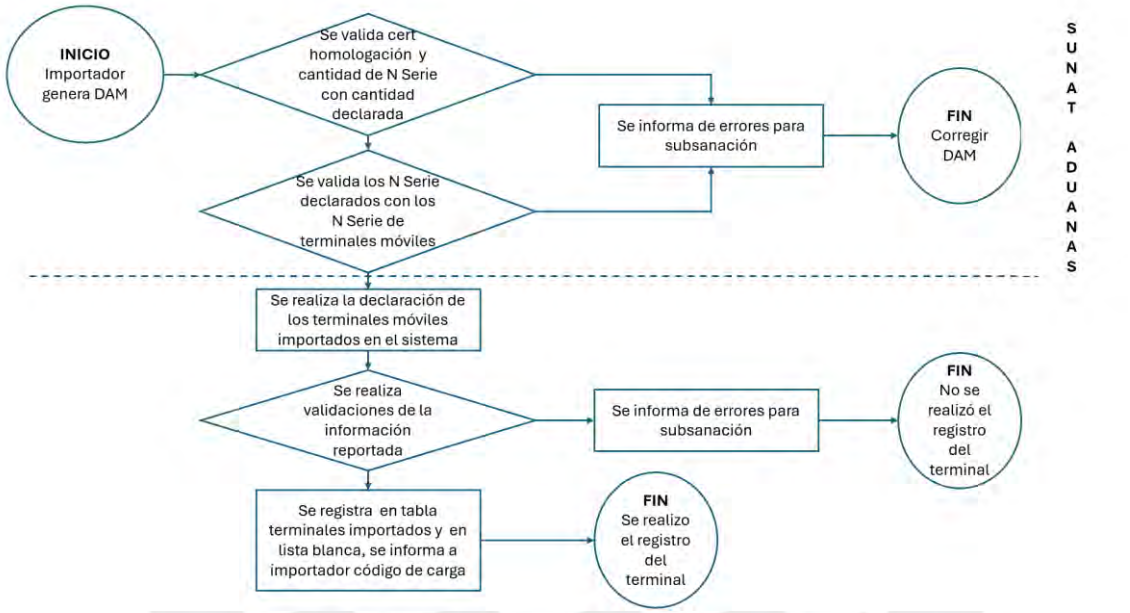


Figura 10: Propuesta de proceso de registro de terminales de terminales importados
Fuente y elaboración propia

Otro aspecto que es necesario abordar es la oportunidad denominada *"Existe una baja tasa de recuperación de terminales móviles robados por parte de las autoridades como MINITER"*. En este contexto según la información publicada por entidades como Ministerio Publico y Policía Nacional, se realizan regularmente operativos en lugares que vendería equipos terminales robados o adulterados, realizándose las validaciones de la correspondencia del IMEI pregrabado con el IMEI lógico y si estos se encuentran registrados en lista negra. Sin embargo, se plantea los siguientes aspectos de mejora:

- Debido a que en el punto anterior se considera también el registro del número de serie de los terminales móviles, las validaciones que realizan las autoridades deben incluir si el

número de serie está registrado en lista blanca y si existe correspondencia con el IMEI.

- Se debe incluir en los operativos los establecimientos comerciales que venden partes de terminales móviles, como pantallas y cámaras; en estos establecimientos las autoridades de la Gerencia de Prevención del Contrabando y Operaciones Especiales de la SUNAT deben revisar la documentación que sustente la importación de las partes que están a la venta. Considerando ello, es necesario que los documentos de importación de partes de terminales móviles incluyan los números de serie.

3.2.1.2 Diseño del proceso de gestión de acceso a la red móvil por lista blanca y lista negra mediante un CEIR sincrónico.

Según el análisis de factores internos realizado en el numeral 3.1.2 del presente trabajo, se identificó las debilidades *"no se registra información que permita generar un proceso que ayude a la recuperación de terminales móviles robados"* y *"la no habilitación de la lista blanca genera que terminales móviles alterados puedan operar en las redes móviles".*

Considerando ello, a continuación, se describen algunas casuísticas que se realizarían por los ladrones luego de realizado un robo:

Caso 1: Equipos con Sim Card físico y/o Batería de fácil retiro.

a) Se ejecuta el robo y el ladrón se trasladaría a un punto cercano para apagar el terminal móvil, habilitar el modo avión o retirar el sim card.
b) El ladrón traslada 1 o varios terminales móviles a un punto, donde intentará vender estos equipos.
c) El receptor de equipos robados intentará encender los equipos o habilitar la conexión para acceder a la información de bancos o contactos para solicitar dinero.
d) El receptor de equipos robados intentará formatear el equipo o configurarlo en modo root en caso no pueda desbloquear el equipo.
e) El receptor de equipos robados validará si el equipo se encuentra bloqueado, intentará adulterar el IMEI e intentará validar si con este IMEI el terminal móvil puede operar, para que pueda ser vendido.

En este caso, es posible que los ladrones intenten encender el equipo y tratar de acceder a información, y en tanto el terminal tenga un sim, generará eventos en la red móvil, aun cuando el servicio móvil del usuario original se encuentre ya suspendido.

Caso 2: Equipos con Esim / Baterías no retirables con facilidad

a) Se ejecuta el robo y el ladrón se trasladaría a un punto cercano para apagar el terminal móvil, habilitar el modo avión o retirar el sim card; sin embargo, al tener un Esim no podrá desvincular el servicio móvil y aun cuando apague el equipo, este seguirá generando eventos en la red móvil.
b) El ladrón traslada una o varios terminales móviles a un punto, donde intentará vender estos equipos.
c) El receptor de equipos robados intentará acceder a la información de bancos o contactos para solicitar dinero.
d) El receptor de equipos robados intentará formatear el equipo o configurarlo en modo root en caso no pueda desbloquear el equipo.
e) El receptor de equipos robados validará si el equipo se encuentra bloqueado, intentará adulterar el IMEI e intentará validar si con este IMEI el terminal móvil puede operar, para que pueda ser vendido.

Bajo el análisis de los casos presentado, es posible afirmar que un terminal móvil genere eventos de location update o IMSI atach, luego de producido un robo; sin embargo, bajo la solución actual no sería posible registrar esta información en el RENTESEG, siendo esta información muy útil para generar un proceso de recuperación de equipos terminales.

Considerando lo antes presentado, la estrategia inicial sería crear una Lista Gris, bajo las siguientes características:

- Los IMEI que no se encuentren en Lista Blanca o en Lista Negra pertenecerán a la lista gris.
- Si bien estos terminales móviles tendrán acceso a la red móvil, todos los eventos de location update e imsi atach serán registrados en la entidad centralizada a fin de generar trazabilidad.
- Los equipos que sean reportados como robados, permanecerán en la lista gris por un periodo de tiempo antes de pasar a la lista negra, de manera que la entidad

centralizada registre los eventos location update e imsi atach posteriores al reporte de robo. El tiempo de permanencia en Lista Gris sería aleatorio.

- Los equipos que se registran en lista gris por no estar en lista blanca, así como, se registrarán los eventos.

Asimismo, es necesario recordar el flujo de información para la gestión de acceso de un CEIR, detallado en el documento ITU-T Q-Sup.76, tal como se muestra a continuación:

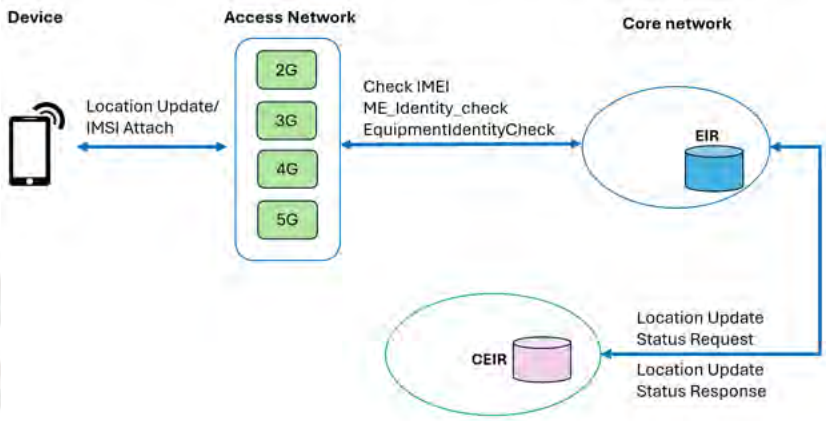


Figura 11: Gestión de acceso a la red móvil
Fuente [7]

Considerando ello, se definen dos procesos según el evento de la red móvil que genera la consulta:

- Respecto del proceso para el evento IMSI Attach, se genera cuando un terminal móvil se enciende o se cambia el Sim Card, por tanto, el proceso considera confirmar o denegar el acceso a la red, registrar los eventos generados. Considerando ello, en la Figura 12, se muestra la propuesta para el proceso de gestión de acceso a la red móvil.

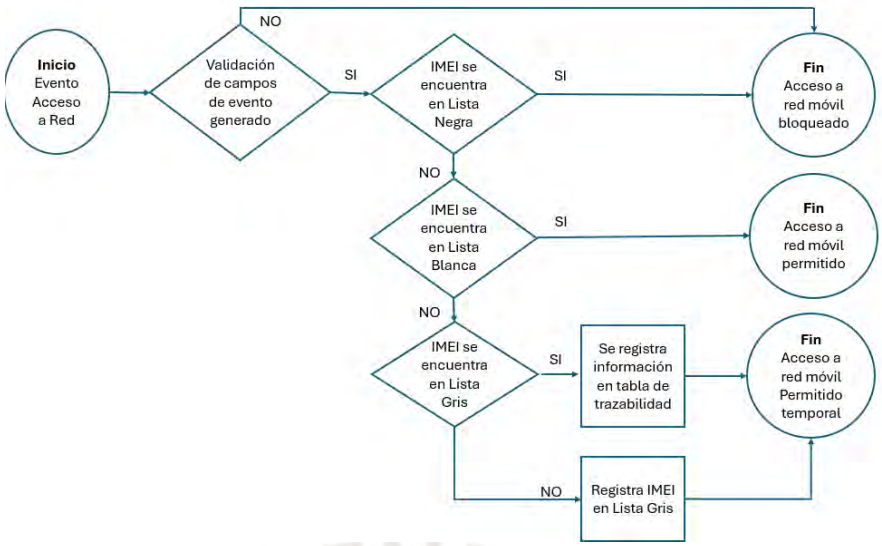


Figura 12: Propuesta de proceso de gestión de acceso a la red móvil
Fuente y elaboración propia

Asimismo, en la Figura 13, se muestra la propuesta del proceso de validación programado diariamente que analiza la información a fin de detectar IMEI inválidos, inoperativos y uso prohibido.

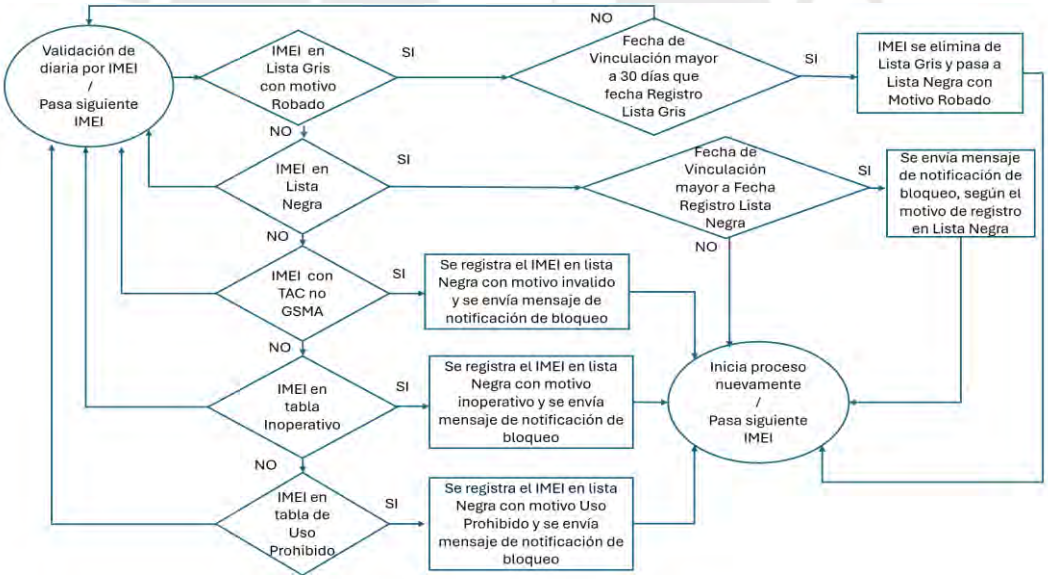


Figura 13: Propuesta de proceso de validación diaria
Fuente y elaboración propia

3.2.1.3 Diseño del proceso de trazabilidad de terminales móviles robados.

Por otro lado, considerando que el proceso de gestión de acceso a la red móvil, se utiliza las tablas actualizadas de lista negra, lista blanca y lista gris, en las Figuras 14 y 15, se

muestran las propuestas de proceso para realizar reportes de robo y reportes de recuperación.

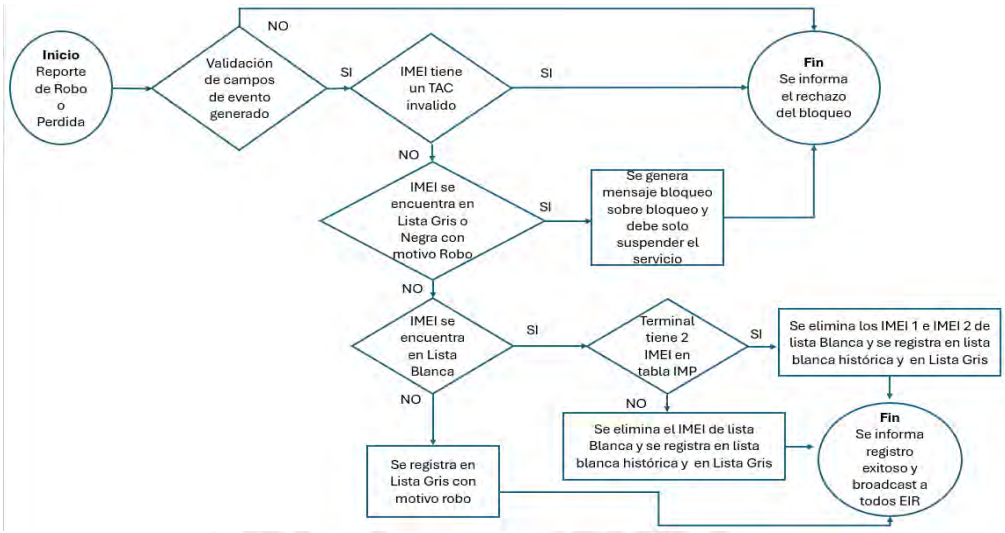


Figura 14: Propuesta de proceso para un reporte de robo
Fuente y elaboración propia

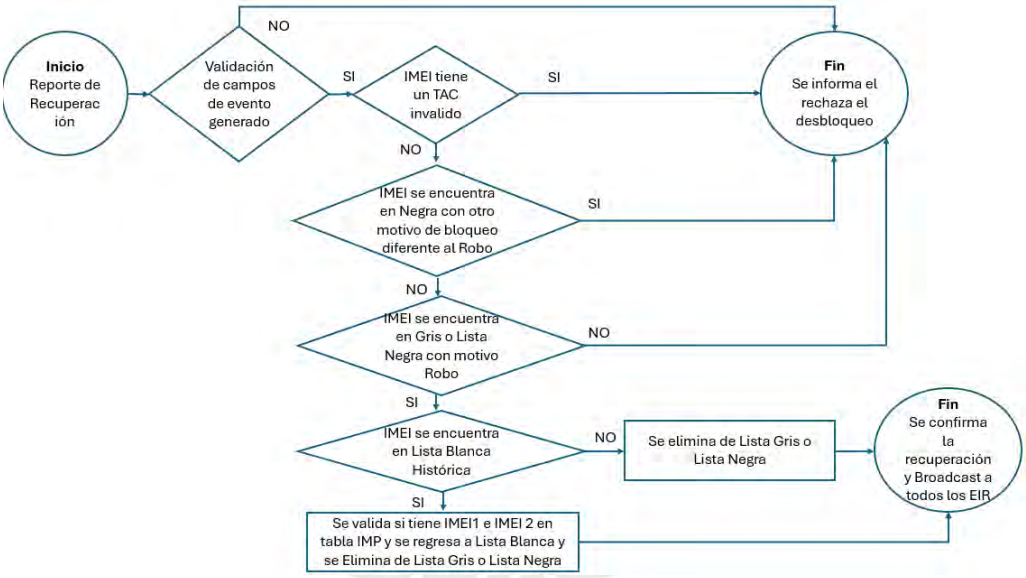


Figura 15: Propuesta de proceso para un reporte de recuperación
Fuente y elaboración propia

3.2.1.4 Modificaciones normativas necesarias

Respecto a la implementación de la Lista Gris, se debe señalar que, en la situación actual, cuando se reporta un robo, el proceso genera que el terminal móvil no se pueda conectar a la red móvil, por lo que no se pueden realizar acciones posteriores sobre el terminal o el servicio móvil asociado. Según las experiencias de otros países [22], el implementar una Lista Gris y los correspondientes procedimientos para la

recuperación de equipos tendría un efecto positivo al incrementar las recuperaciones.

Ajuste del marco normativo, respecto al registro de terminales adquiridos en el extranjero, en el extremo de que exista un control aduanero y el proceso de registro pueda realizarse mediante herramientas informáticas sin la necesidad de trámites presenciales.

Finalmente, el marco normativo actual contempla que los concesionarios móviles remiten la información de las vinculaciones realizadas por todos los servicios móviles y la evaluación de los terminales que pueden o no acceder a la red se realiza posterior a 24 horas, generando que, cuando un usuario compra un equipo y lo prueba, no tiene certeza de que no tendrá problemas posteriormente, por tanto se recomienda el uso de la gestión de acceso a la red móvil mediante el modelo de CEIR de API sincrónico. Considerando ello, la modificación normativa debe considerar que los eventos de vinculación ya no se realizan diariamente, a partir del cambio normativo se realizarán consultas en tiempo real al CEIR para los casos de equipos terminales que se detecten por primera vez en la red de concesionario móvil. El cambio normativo deberá estar alineado a las recomendaciones del documento ITU-T Q-Sup.76[7] a fin que se establezcan las características técnicas necesarias para la operación del CEIR sincrónico.

3.2.2 Diseño de la arquitectura de la solución

Considerando que el Perú existen más de 40 millones de servicios móviles, y que la propuesta del diseño del proceso de gestión a la red móvil considera que las peticiones se realicen el tiempo real, los componentes de la solución deben considerar las capacidades y configuraciones que permitan operar y brindar respuesta en el tiempo máximo establecido, asimismo debe considerar la flexibilidad para poder brindar mayores recursos considerando las diferentes etapas en la que puede operar el sistema, por ejemplo:

- Etapa de desarrollo
- Etapa de pruebas internas
- Etapa de pruebas con concesionarios
- Etapa de puesta a producción en fases
- Etapa de operación

Considerando la flexibilidad en la capacidad de los recursos, la propuesta de diseño considera que la solución debe implementarse sobre la nube considerando el modelo SaaS. Asimismo, para la etapa de diseño en la que se enmarca la propuesta de solución, se selecciona los servicios SaaS de AWS al brindar la opción de poder desplegar servicios gratuitos durante 12 meses.

3.2.2.1 Herramientas fundamentales

Los principales componentes de la arquitectura para la solución tecnológica que brinda AWS como servicios SaaS son:

Amazon Relational Database Service (Amazon RDS) Solución de base de datos relacional que automatiza las tareas de administración (despliegue, respaldo y aplicación de parches) [17].

AWS Lambda: servicio informático sin servidor y de auto activación a demanda, utilizado para el procesamiento de información remitida por dispositivos para su atención y respuesta [17].

Amazon Api Gateway: servicio proporcionado por Amazon, destinado a los desarrolladores para la creación, publicación, mantenimiento, monitoreo y protección de APIs a cualquier escala [17].

3.2.2.2 Configuración de Base de datos

Respecto a la configuración de la base de datos, se tiene 2 opciones de motor de base datos disponibles en la función Amazon RDS, MySQL y PostgreSQL.

Según la recomendación de AWS, para elegir entre PostgreSQL o MySQL, respecto al ámbito de aplicación, se tiene lo siguiente:

- PostgreSQL utilizado para operaciones de escritura frecuentes y consultas complejas, así como para cubrir necesidades empresariales.

- Sin embargo, puede iniciarse un proyecto de MySQL si la intención del desarrollador es crear un modelo.

En ese sentido, para la propuesta de diseño se utilizará el motor de base de datos MySQL, siguiendo las

recomendaciones de AWS y considerando el nivel gratuito para la etapa de diseño. En las Figuras 16 y 17, se puede observar la base de datos habilitada en la consola de administración de AWS y las características de almacenamiento.

Figura 16: Resumen de consola de AWS de base de datos habilitada

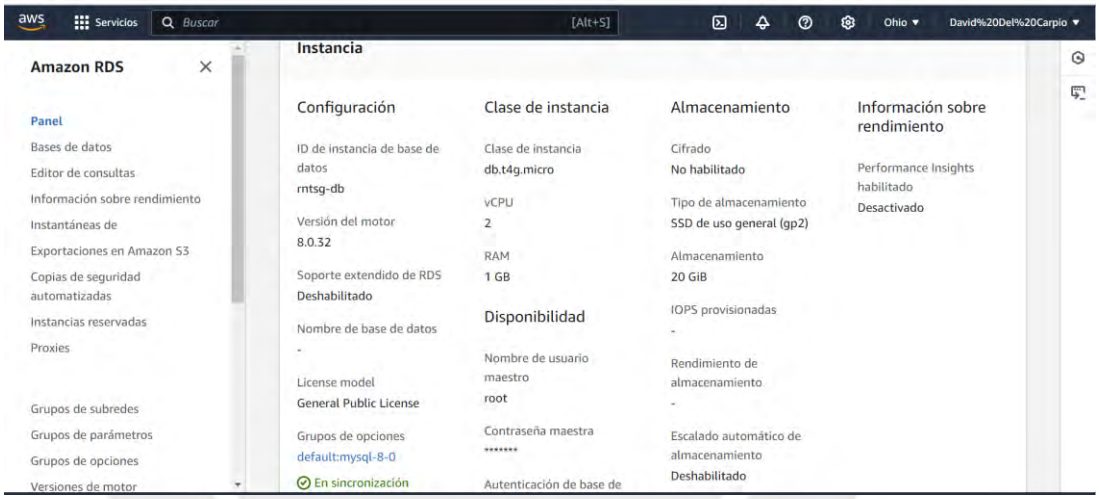


Figura 17: Configuración de la base de datos habilitada en AWS

3.2.2.3 Configuración de servicios de intercambio de información

Tal como se detalló previamente, utilizaremos el servicio Lambda AWS para optimizar los recursos y utilizar los servicios que se requieran a demanda, siendo que en cada solicitud de acceso al servicio Lambda desplegará un servidor para la ejecución correspondiente del código fuente definido para el servicio.

Figura 18: Servicios desplegados en Lambda AWS

Asimismo, para acceder a los servicios desplegados en Lambda, es necesario configurar el servicio API Gateway para que estos servicios sean accedidos desde Internet. En la Figura 18, se muestra el diagrama generado por AWS luego de la asociación del API Gateway con los servicios configurados en Lambda.

Figura 19: Diagrama de conexión generado en consola AWS

Por otro lado, para acceder a los servicios proporcionados por las API, el API Gateway define el punto de enlace:

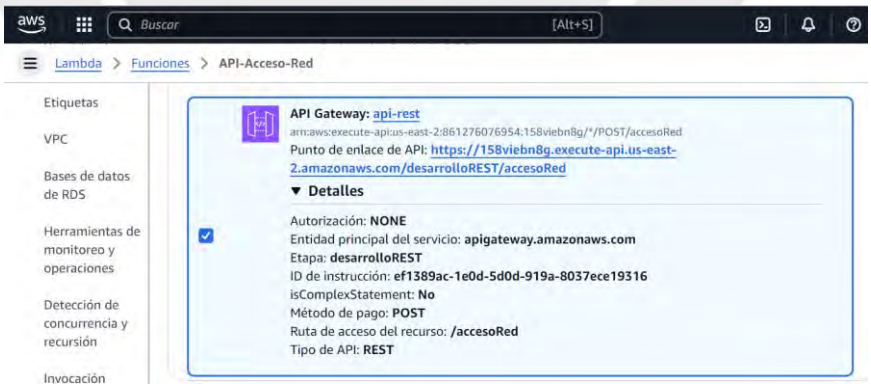


Figura 20: Detalle de configuración API Gateway AWS

Finalmente, el código fuente que se ejecutará ante una petición o intercambio de información con el CEIR se encontrará desplegado como una función en el servicio Lambda AWS. Como se puede visualizar en las Figuras 21 y 22.

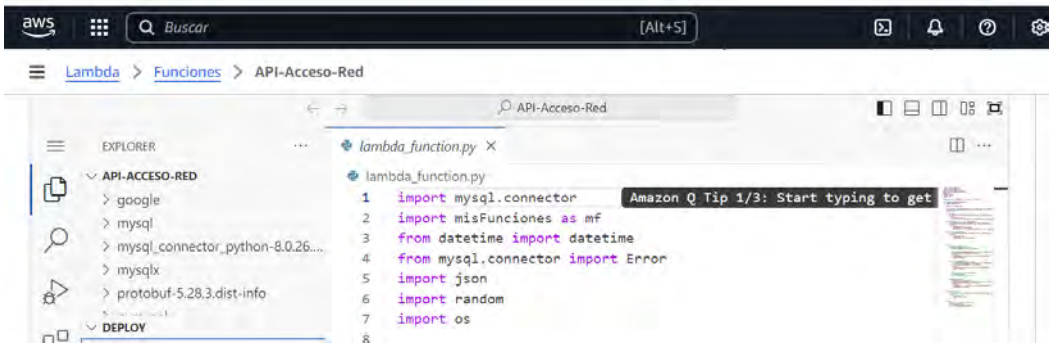


Figura 21: Código fuente como función de Lambda AWS

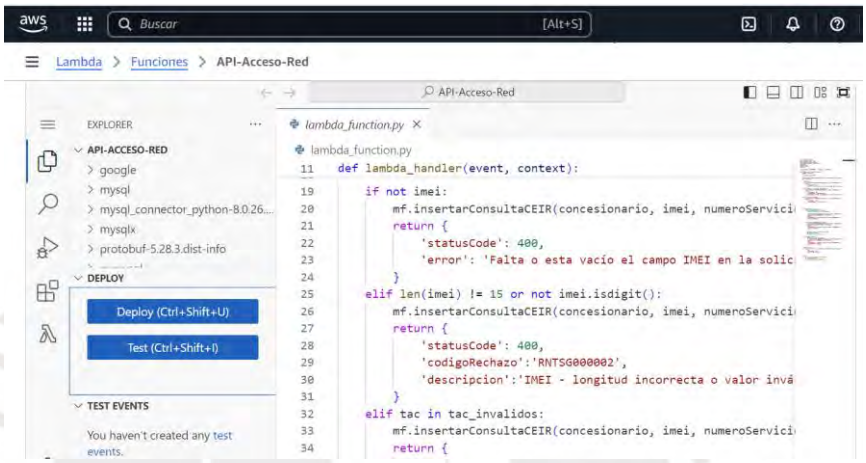


Figura 22: Continuación código fuente como función de Lambda AWS

3.2.3 Diseño del modelo de las bases de datos

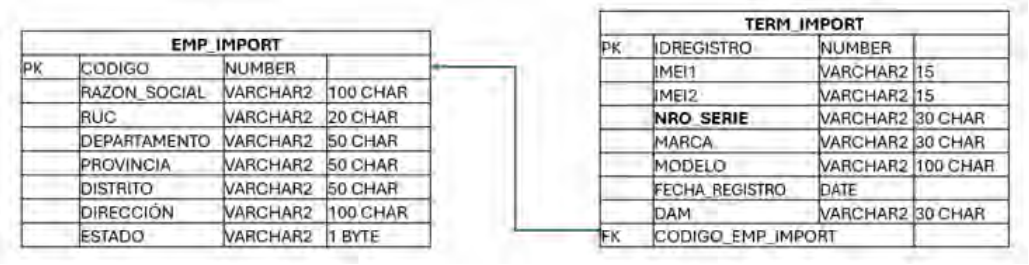
Considerando los requerimientos de respuesta en tiempo real y las características de big data para el tratamiento de la información, se detalla el modelo de base de datos para la propuesta de solución tecnológica.

3.2.3.1 Modelo de la base de datos de lista blanca

Considerando el proceso para el registro de terminales importados, el modelo incluye la tabla maestra “EMP_IMPORT”, en la cual se registrarán todas las empresas importadoras que importan terminales móviles. La información es declarada ante SUNAT. Cabe precisar que, cuando se acepta el registro de una empresa se asigna un identificador único en el campo “CODIGO”, asimismo se define este campo como llave primaria.

Adicionalmente, para que las empresas importadoras registren la información de todos los terminales que importan se define la tabla “TERM_IMPORT”, asignándose a cada nuevo registro un identificador único en el campo

“IDREGISTRO” (Llave primaria -PK) y una llave foránea que identifica la empresa que realizó el registro del terminal con el campo “CODIGO_EMP_IMPORT”. El modelo relacional de las tablas señaladas se muestra en la Figura 23.



EMP_IMPORT			
PK	CODIGO	NUMBER	
	RAZON_SOCIAL	VARCHAR2	100 CHAR
	RUC	VARCHAR2	20 CHAR
	DEPARTAMENTO	VARCHAR2	50 CHAR
	PROVINCIA	VARCHAR2	50 CHAR
	DISTRITO	VARCHAR2	50 CHAR
	DIRECCIÓN	VARCHAR2	100 CHAR
	ESTADO	VARCHAR2	1 BYTE

TERM_IMPORT			
PK	IDREGISTRO	NUMBER	
	IMEI1	VARCHAR2	15
	IMEI2	VARCHAR2	15
	NRO_SERIE	VARCHAR2	30 CHAR
	MARCA	VARCHAR2	30 CHAR
	MODELO	VARCHAR2	100 CHAR
	FECHA_REGISTRO	DATE	
	DAM	VARCHAR2	30 CHAR
FK	CODIGO_EMP_IMPORT		

Figura 23: propuesta de Modelo relacional para Registro Terminales Importados
Fuente y elaboración propia

Asimismo, para el proceso de registro de terminales adquiridos en el extranjero, se define la tabla maestra “USU_ADUANA” en esta tabla se registrarán todos los funcionarios de aduanas que se encuentren designados en un punto de control migratorio, asignándose el identificador único campo “CODIGO”.

Además, como se señaló en la definición del proceso de registro de terminales adquiridos en el extranjero, cuando una persona en un punto de control migratorio quiere declarar el ingreso de un terminal, el funcionario de aduanas registra la información y esta se registrará en la tabla “TERMINAL_DECLA”, asignándose también un código de verificación OTP, para que pueda activar la aplicación móvil que permita realizar el registro.

Finalmente, se considera la tabla operativa “REGISTRO_ADQUIRIDOS_EXT”, la cual almacenará la información de los terminales móviles que completaron el proceso de registro a través de la aplicación móvil asignándole un identificador único “IDREGISTRO”. El modelo relacional de las tablas señaladas se muestra en la Figura 24.

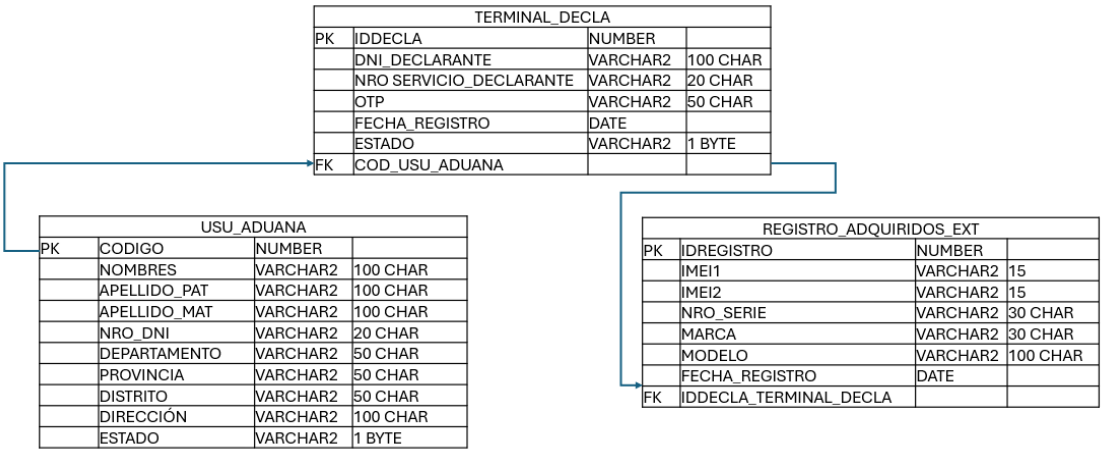


Figura 24: Modelo relacional para el registro terminales adquiridos en el extranjero
Fuente y elaboración propia

Finalmente, es necesario considerar que los terminales que se registren válidamente en las tablas “TERM_IMPORT” y “REGISTRO_ADQUIRIDOS_EXT”, pasarán a la tabla “LISTA_BLANCA”.

3.2.3.2 Modelo de base de datos lista negra, lista gris y lista blanca

La información de los reportes de robo y recuperación se registran en la tabla denominada “MSG_ROBO_RECUP” considerando como identificador único el campo “IDMSG”. Si la información corresponde a un reporte de robo y no genera un rechazo, se registra la información en la tabla “LISTA_GRIS” y se elimina la información de la tabla “LISTA_BLANCA”, registrando la eliminación en la tabla “LISTA_BLANCA_HIST”. Los IMEI permanecen en esta lista por 15 días y luego de ello pasan a la tabla “LISTA_NEGRA”.

Asimismo, si los IMEI son reportados como recuperados, regresan a la tabla “LISTA_BLANCA”, siempre que existan previamente en la tabla “LISTA_BLANCA_HIST”. El modelo relacional de las tablas señaladas se muestra en la Figura 25.

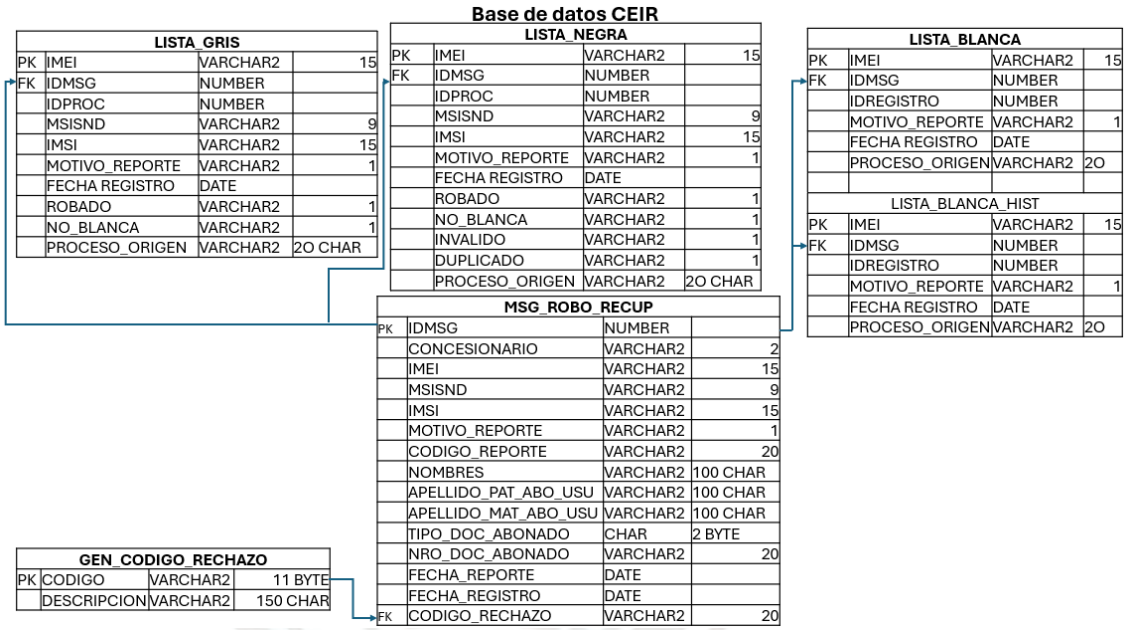


Figura 25: Modelo relacional de lista negra, lista gris y lista blanca
Fuente y elaboración propia

3.2.3.3 Modelo de la base de datos de trazabilidad

Considerando que los eventos de acceso a la red llegan a la base de datos para realizar la validación del estado de un IMEI, consultando si este se encuentra en lista blanca, lista negra o se encuentra en lista gris, los eventos se registrarán en la base de datos de trazabilidad, en la tabla "CONSULTAS_CEIR", en caso la información no cumpla con alguna de las validaciones se generará un "CODIGO_RECHAZO". El modelo relacional de las tablas señaladas se muestra en la Figura 26.

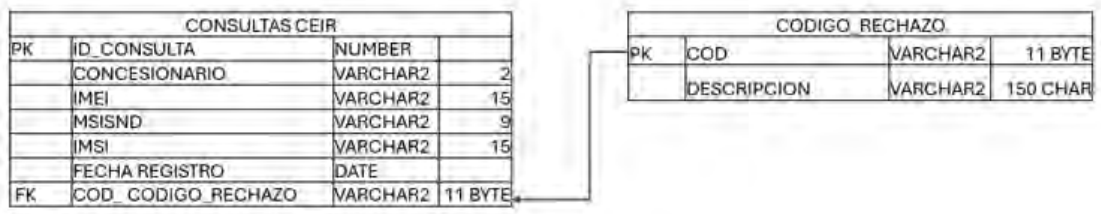


Figura 26: Propuesta de modelo relacional para consulta CEIR
Fuente y elaboración propia

3.2.4 Diseño de los servicios para la solución

Antes de realizar el diseño específico del flujo de información, se define la arquitectura de la solución, tal como se muestra en la Figura 27.

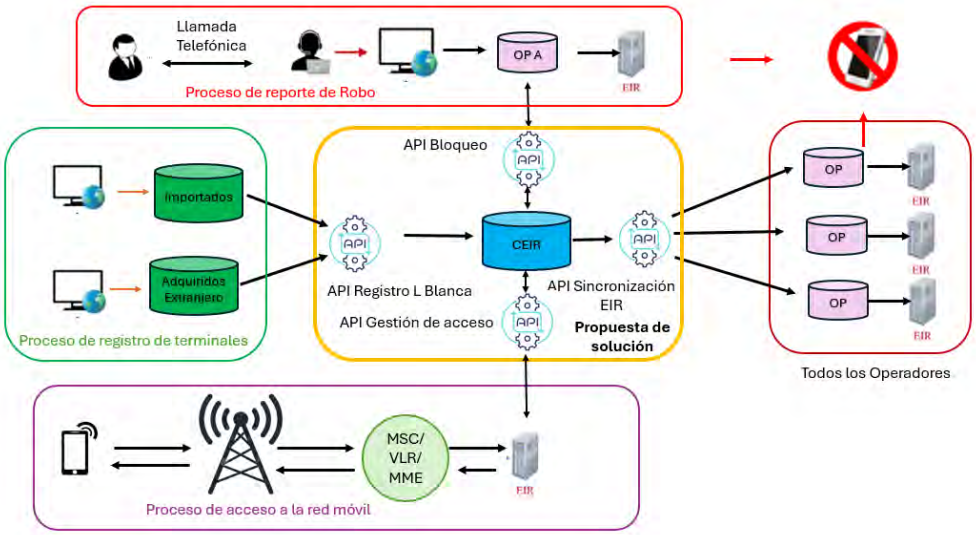


Figura 27: Arquitectura para la propuesta de solución
Fuente y elaboración propia

El orden para la definición de los flujos de intercambio de información se define considerando la dependencia de la información para cada uno de los flujos. Considerando ello el orden para la definición del diseño será el siguiente:

a) API Bloqueo / Desbloqueo: Este servicio permitirá que los concesionarios móviles reporten al CEIR el bloqueo o desbloqueo de un equipo terminal.
b) API Registro Lista Blanca: Este servicio permitirá registrar información en la lista blanca una vez que se complete los procesos de registro de terminales importados o de terminales adquiridos en el extranjero.
c) API Sincronización de EIR: Este servicio se utilizará en los siguientes casos:
   c.1 Compartir a todos los concesionarios un mensaje para informar un reporte de robo realizado.
   c.2 Compartir a todos los concesionarios un mensaje para informar un reporte de recuperación realizado.
   c.3 Compartir a todos los concesionarios mediante mensajes el resultado del proceso de validación diaria.
d) API Gestión de acceso a la Red: Este servicio se utiliza cada vez que se genere un evento de IMSI Attach o Location Update en las redes de los concesionarios.

3.2.4.1 Diseño del flujo de intercambio de información para API Bloqueo / Desbloqueo

En caso de un reporte de robo, el proceso contempla realizar las siguientes acciones secuencialmente:

Paso 1: Inicia cuando el Concesionario (MNO) realiza una solicitud al CEIR para ejecutar el registro de robo de un terminal.

Paso 2: El CEIR recibe mediante el API Bloqueo / Desbloqueo la solicitud de registro, y realiza las siguientes validaciones:

2.1 Existe IMEI en tabla Lista_Negra, si encuentra el IMEI el API responde con un mensaje de error “IMEI ya registrado en Lista Gris / Lista Negra”.

2.2 Existe IMEI en tabla Lista_Blanca, si encuentra el IMEI, consultará si tiene un IMEI1 o IMEI2 asociado; se eliminará los IMEI de lista blanca y se guarda el registro de eliminación en la tabla Lista_Blanca_Hist, se registrará el IMEI en Lista_Gris y el API responderá con un mensaje exitoso. Adicionalmente, se realizará una llamada al API Sincronización del EIR para remitir el mensaje broadcast de bloqueo "Elimina de permetted List".

2.3 Existe IMEI en tabla Lista Gris, si encuentra el IMEI, el API responde con un código de error “IMEI ya registrado en Lista Gris”.

2.4 Si no se encuentra el IMEI en ninguna de las Listas, se registra el IMEI en Lista Gris y el API responderá con un mensaje exitoso.

Es necesario considerar que, transcurridos una cantidad aleatoria de días, un IMEI en Lista Gris pasa a Lista Negra en el CEIR, y se realiza una llamada al API Sincronización del EIR "Registra en Block List", realizado el bloqueo del IMEI y la suspensión de los servicios móviles que generaron vinculación posterior al reporte de robo. Los abonados podrán solicitar la activación del servicio en la empresa operadora presentando el documento entregado por la Policía luego de realizar la devolución del terminal móvil.

A fin de determinar los casos de prueba para la validación del diseño del flujo definido, se establecen los siguientes casos de uso:

- IMEI reportado registrado en Lista Negra.
- IMEI reportado registrado en Lista Blanca.

- IMEI reportado registrado en Lista Gris.
- IMEI reportado no registrado en ninguna de las Listas del CEIR.

Asimismo, para definir claramente el comportamiento del API Bloqueo / Desbloqueo se define la estructura del mensaje de entrada para reporte de robo. Para ello se muestra un ejemplo:


```
{
  "concesionario":"28",
  "imei":"313131310000000",
  "numeroServicio":"944444444",
  "imsi":"716440000000000",
  "motivoReporte":"R",
  "codigoReporte":"P00002",
  "nombres":"Armando",
  "apellidoPaterno":"Lambda",
  "apellidoMaterno":"Mercado",
  "tipoDocumento":"1",
  "numeroDocumento":"44444444",
  "fechaReporte":"2024-11-04"
}
```


Figura 28: Propuesta de mensaje JSON de entrada reporte de robo
Fuente y elaboración propia

A continuación, se detallarán los diagramas de estado para cada uno de los casos de uso para ejecución de bloqueos o desbloqueos:

Caso de Uso A-1	
Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado registrado en Lista Negra.

Entrada:	Reporte de Robo
	Mensaje con la siguiente estructura, considerando todos los campos obligatorios:
	{
	"concesionario":"", "imei":"",
	"numeroServicio":"", "imsi":"",
	"motivoReporte":"S", "codigoReporte":"",
	"nombres":"", "apellidoPaterno":"",
	"apellidoMaterno":"", "tipoDocumento":"",
	"numeroDocumento":"", "fechaReporte":""
	}

Acción:	No Aplica
Salida:	Mensaje de error “IMEI ya registrado en Lista Negra” con la siguiente estructura: { "fechaMensaje": "YYYYMMDDHH24mmss", "codigoRechazo": " RNTSG000001"}

Considerando los detalles de caso de uso el diagrama de estado se muestra en la Figura 29.

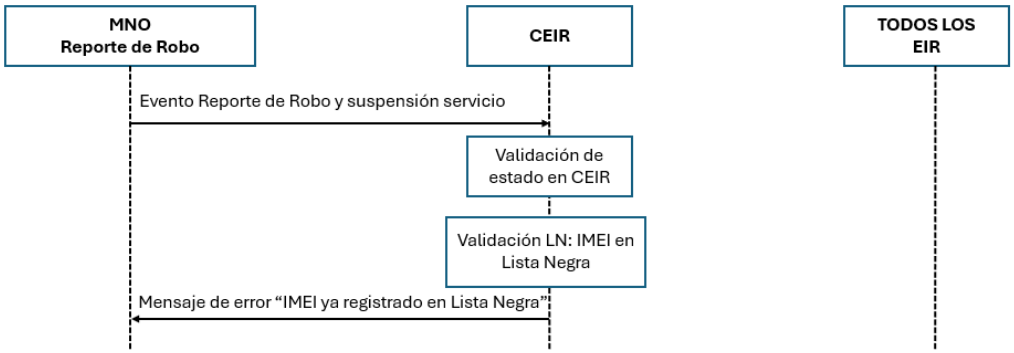


Figura 29: Propuesta de diagrama de estado para el caso de uso A-1
Fuente y elaboración propia

Caso de Uso A-2	
Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado **no** registrado en Lista Negra. IMEI reportado registrado en Lista Blanca.

Entrada:	Reporte de Robo
	Mensaje con la siguiente estructura:
	{
	"concesionario":"", "imei":"",
	"numeroServicio":"", "imsi":"",
	"motivoReporte":"S", "codigoReporte":"",
	"nombres":"", "apellidoPaterno":"",
	"apellidoMaterno":"", "tipoDocumento":"",
	"numeroDocumento":"", "fechaReporte":""
	}

Acciones:	- Eliminar IMEI de lista blanca. - Guardar el registro de eliminación en la tabla Lista_Blanca_Hist. - Registrar el IMEI en Lista_Gris.

Salida 1:	Mensaje registro exitoso con la siguiente estructura:
	{
	"fechaMensaje": "YYYYMMDDHH24mmss",
	"Resultado": "Sustraído",
	"codigoRechazo": ""
	}

Salida 2:	Recurso: API Sincronización del EIR Mensaje broadcast de bloqueo "Elimina de permetted List".

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 30.

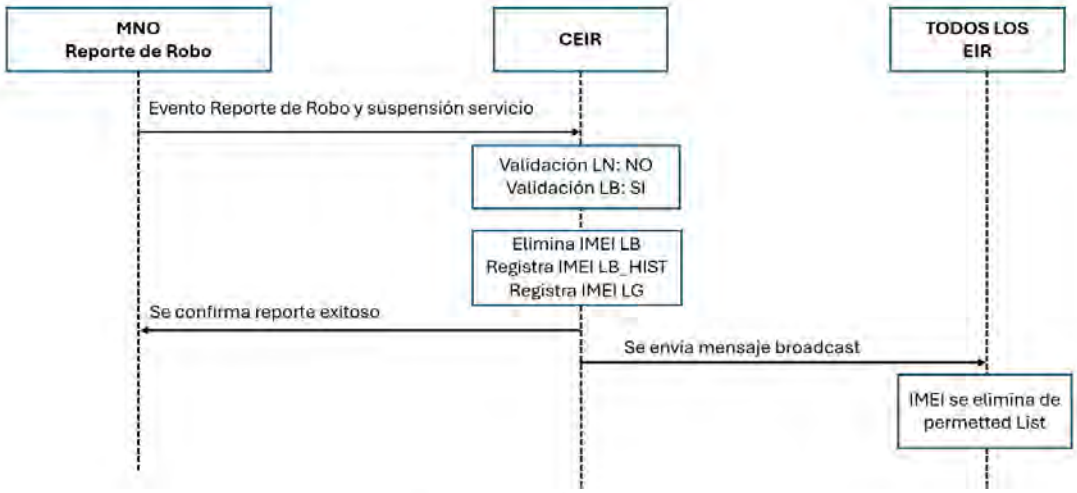


Figura 30: Propuesta de diagrama de estado para el caso de uso A-2
Fuente y elaboración propia

Caso de Uso A-3	
Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado **no** registrado en Lista Negra. IMEI reportado **no** registrado en Lista Blanca. IMEI reportado registrado en Lista Gris

Entrada:	Reporte de Robo
	Mensaje con la siguiente estructura:
	{
	"concesionario":"", "imei":"",
	"numeroServicio":"", "imsi":"",
	"motivoReporte":"S", "codigoReporte":"",
	"nombres":"", "apellidoPaterno":"",
	"apellidoMaterno":"", "tipoDocumento":"",
	"numeroDocumento":"", "fechaReporte":""
	}

Acciones:	No Aplica

Salida:	Mensaje de error “IMEI ya registrado en Lista Gris” con la siguiente estructura:
	{
	"fechaMensaje": "YYYYMMDDHH24mmss",
	“Resultado”: “Error”,
	"codigoRechazo": "RNTSG000001"
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 31.

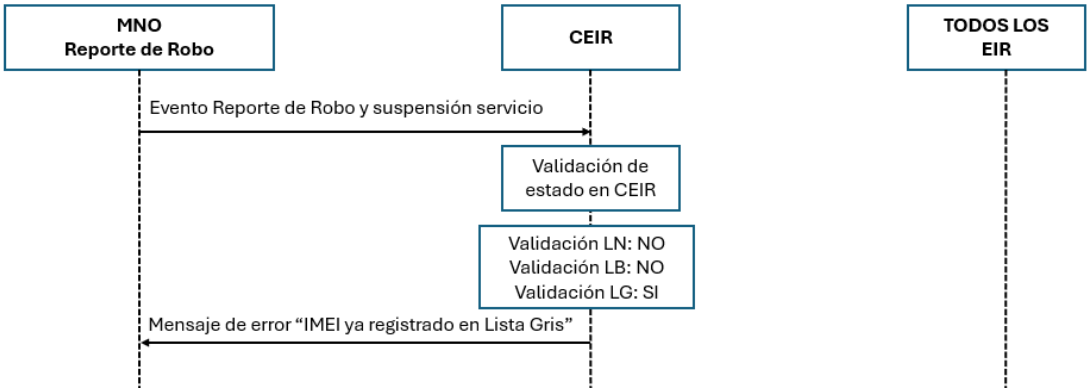


Figura 31: Propuesta de diagrama de estado para el caso de uso 3
Fuente y elaboración propia

Caso de Uso A-4	
Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado **no** registrado en Lista Negra. IMEI reportado **no** registrado en Lista Blanca. IMEI reportado **no** registrado en Lista Gris

Entrada:	Reporte de Robo
	Mensaje con la siguiente estructura:
	{
	"concesionario":"", "imei":"",
	"numeroServicio":"", "imsi":"",
	"motivoReporte":"S", "codigoReporte":"",
	"nombres":"", "apellidoPaterno":"",
	"apellidoMaterno":"", "tipoDocumento":"",
	"numeroDocumento":"", "fechaReporte":""
	}

Acciones:	- Registra el IMEI en Lista Gris

Salida:	Mensaje registro exitoso con la siguiente estructura:
	{
	"fechaMensaje": "YYYYMMDDHH24mmss",
	“Resultado”: “Sustraído”,
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 32.

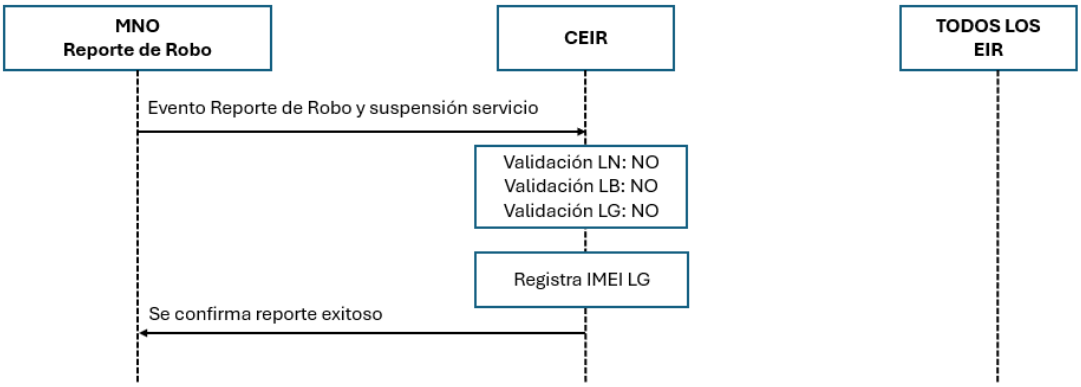


Figura 32: Propuesta de diagrama de estado para el caso de uso A-4
Fuente y elaboración propia

Caso de Uso A-5	
Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado registrado en Lista Negra.
Entrada:	Reporte de Recuperación Mensaje con la siguiente estructura: { "concesionario":"", "imei":"", "numeroServicio":"", "imsi":"", **"motivoReporte":"R"**, "codigoReporte":"", "nombres":"", "apellidoPaterno":"", "apellidoMaterno":"", "tipoDocumento":"", "numeroDocumento":"", "fechaReporte":""}
Acciones:	- Retorna IMEI a lista blanca siempre que se encuentre el IMEI en Lista_Blanca_Hist, si no se encuentra (Excepción 1: Genera Error) - Elimina IMEI de Lista Negra
Salida:	Mensaje registro exitoso con la siguiente estructura: {"fechaMensaje": "YYYYMMDDHH24mmss", “Resultado”: “Recuperado”, "codigoRechazo": "" }
Salida 2:	Recurso: API Sincronización del EIR Mensaje broadcast de bloqueo "Elimina de block list" y registro a “permetted List”.

Considerando los detalles de caso de uso el diagrama de estado se muestra en la Figura 33:

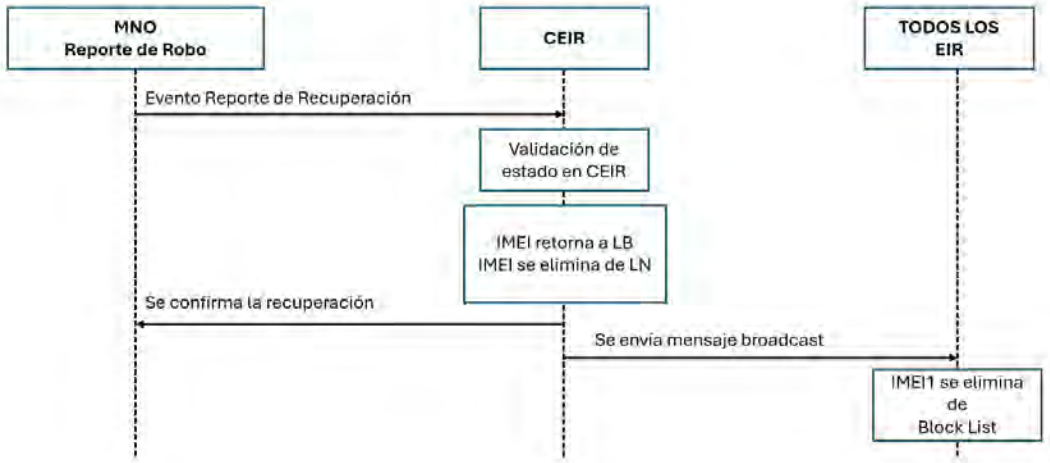


Figura 33: Propuesta de diagrama de estado para el caso de uso A-5
Fuente y elaboración propia

Caso de Uso A-6	
Recurso:	API Sincronización del EIR
Precondición:	IMEI registrado en Lista Gris con una antigüedad aleatoria de días.
Entrada:	No Aplica
Acciones:	- Eliminar IMEI de Lista Gris - Registrar IMEI en Lista Negra
Salida:	Mensaje broadcast de registro "block list" y mensajes de suspensión de todos los servicios que generaron vinculación posterior al reporte de robo.

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 34.

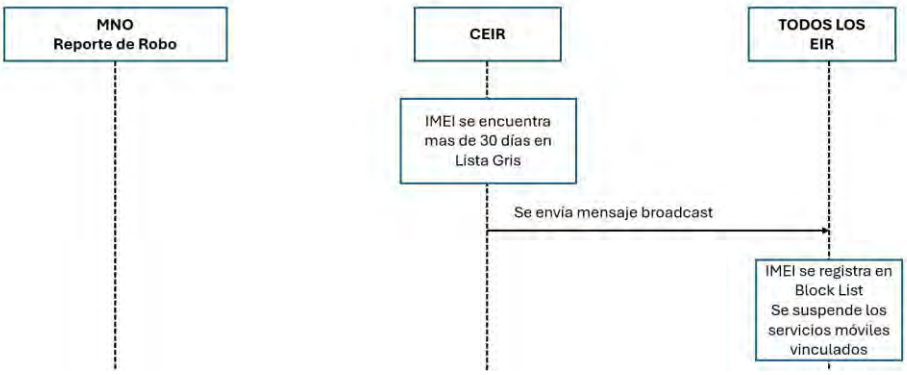


Figura 34: Propuesta de diagrama de estado para el caso de uso A-6
Fuente y elaboración propia

3.2.4.2 Diseño del flujo de intercambio de información para API Registro Lista Blanca

Luego de que los procesos de registro de terminales importados y registro de terminales adquiridos en el extranjero

se completan exitosamente, estas bases de datos intercambiarán información con la base de datos del CEIR mediante el API Registro Lista Blanca. A fin de definir claramente el comportamiento del API Registro Lista Blanca, es necesario definir la estructura del mensaje de entrada para intento de registro, para ello se muestra un ejemplo:

```
{
"idMsg": "11111",
"idRegistro": "44444",
"imei":"777777770000000",
"fechaRegistro":"2024-11-04",
"proceso_origen":"Importado(*)"
}
(*)El proceso Origen puede ser tambien AdqExtr
```

Asimismo, se definen los siguientes casos de uso:

Caso de Uso B-1	
Recurso:	API Registro Lista Blanca
Precondición:	IMEI registrado en Lista Negra

Entrada:	Registro de terminal importado o terminal adquirido en el extranjero.
	Mensaje con la siguiente estructura:
	{
	"idMsg":"", "idRegistro":"", "imei":"",
	"fechaRegistro":"", "proceso_origen":""
	}

Acciones:	No Aplica

Salida:	Mensaje registro rechazado “IMEI registrado en Lista Negra” con la siguiente estructura:
	{
	“idRegistro”: “”,
	"fechaMensaje": "YYYYMMDDHH24mmss",
	“Resultado”: “Error”,
	"codigoRechazo": "APIREGLB001"
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 35:

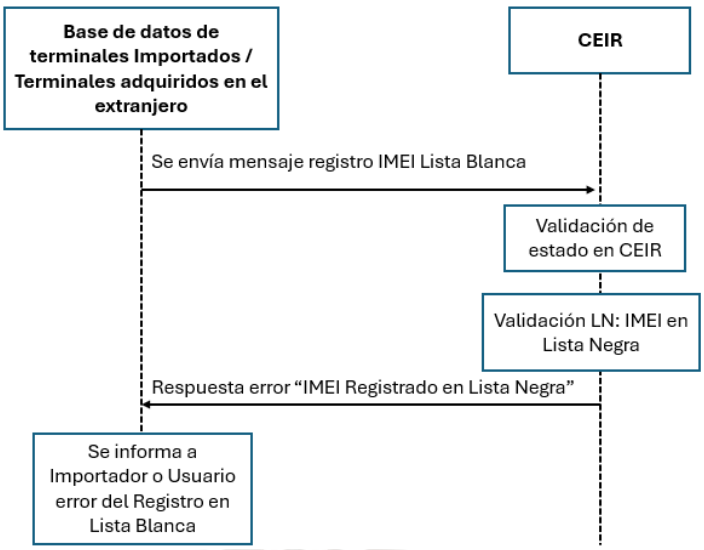


Figura 35: Propuesta de diagrama de estado para el caso de uso B-1
Fuente y elaboración propia

Caso de Uso B-2	
Recurso:	API Registro Lista Blanca
Precondición:	IMEI **no** registrado en Lista Negra IMEI registrado en Lista Gris como Robado

Entrada:	Registro de terminal importado o terminal adquirido en el extranjero.
	Mensaje con la siguiente estructura:
	{
	"idMsg":"", "idRegistro":"", "imei":"",
	"fechaRegistro":"", "proceso_origen":""
	}

Acciones:	No Aplica

Salida:	Mensaje registro rechazado "IMEI reportado como robado" con la siguiente estructura:
	{
	"idRegistro": "",
	"fechaMensaje": "YYYYMMDDHH24mmss",
	"Resultado": "Error",
	"codigoRechazo": "APIREGLB002"
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 36:

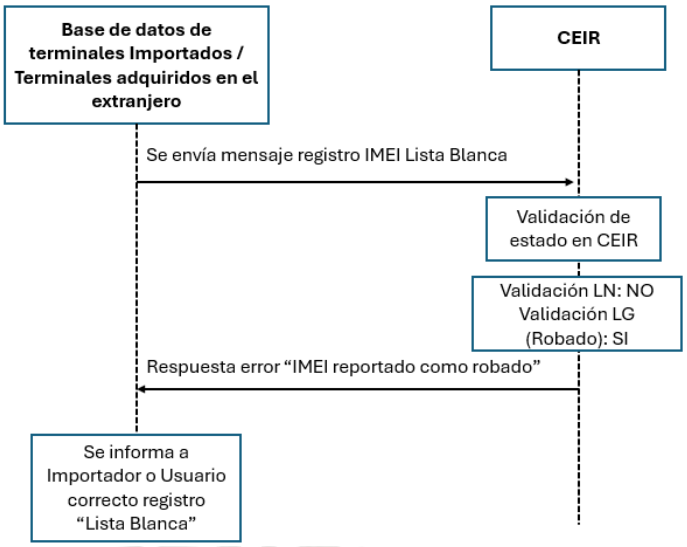


Figura 36: Propuesta de diagrama de estado para el caso de uso B-2
Fuente y elaboración propia

Caso de Uso B-3	
Recurso:	API Registro Lista Blanca
Precondición:	IMEI no registrado en Lista Negra IMEI registrado en Lista Gris como no registrado en Lista Blanca

Entrada:	Registro de terminal importado o terminal adquirido en el extranjero.
	Mensaje con la siguiente estructura:
	{
	"idMsg":"", "idRegistro":"", "imei":"",
	"fechaRegistro":"", "proceso_origen":""
	}

Acciones:	Se elimina el IMEI de Lista Gris Se registra el IMEI en Lista Blanca

Salida:	Mensaje registro exitoso "Se regularizo el registro del IMEI" con la siguiente estructura:
	{
	"idRegistro": "",
	"fechaMensaje": "YYYYMMDDHH24mmss",
	"Resultado": "Regularizado",
	"codigoRechazo": ""
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 37:

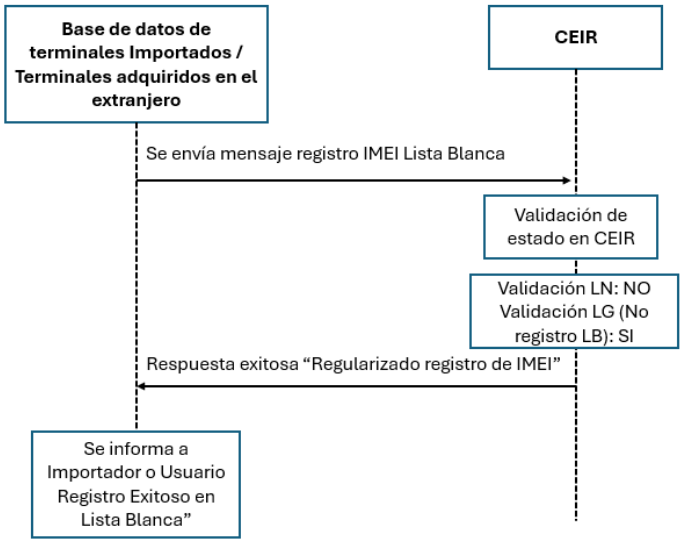


Figura 37: Propuesta de diagrama de estado para el caso de uso B-3
Fuente y elaboración propia

Caso de Uso B-4	
Recurso:	API Registro Lista Blanca
Precondición:	IMEI **no** registrado en Lista Negra. IMEI **no** registrado en Lista Gris. IMEI registrado en Lista Blanca

Entrada:	Registro de terminal importado o terminal adquirido en el extranjero.
	Mensaje con la siguiente estructura:
	{
	"idMsg":"", "idRegistro":"", "imei":"",
	"fechaRegistro":"", "proceso_origen":""
	}

Acciones:	No Aplica

Salida:	Mensaje registro rechazado "IMEI previamente registrado en Lista Blanca" con la siguiente estructura:
	{
	"idRegistro": "",
	"fechaMensaje": "YYYYMMDDHH24mmss",
	"Resultado": "Error",
	"codigoRechazo": "APIREGLB003"
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 38:

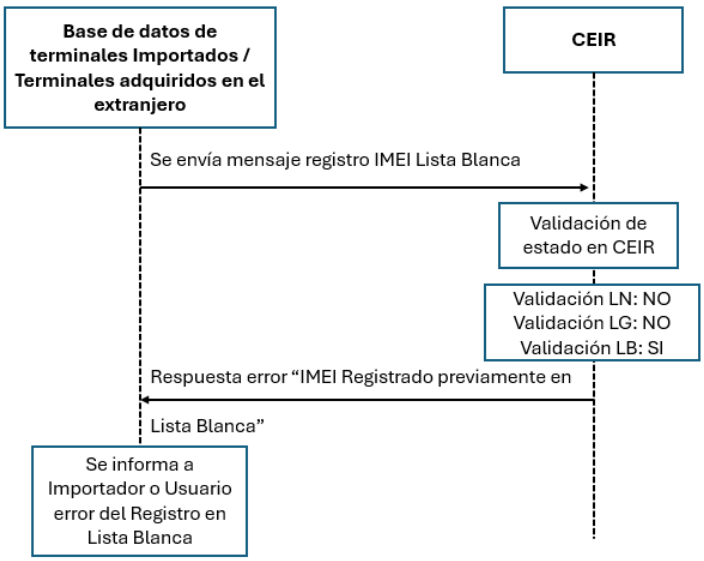


Figura 38: Propuesta de diagrama de estado para el caso de uso B-4
Fuente y elaboración propia

Caso de Uso B-5	
Recurso:	API Registro Lista Blanca
Precondición:	IMEI **no** registrado en Lista Negra IMEI **no** registrado en Lista Gris IMEI **no** registrado en Lista Blanca

Entrada:	Registro de terminal importado o terminal adquirido en el extranjero.
	Mensaje con la siguiente estructura:
	{
	"idMsg":"", "idRegistro":"", "imei":"",
	"fechaRegistro":"", "proceso_origen":""
	}

Acciones:	No Aplica

Salida:	Mensaje registro exitoso "IMEI se registró en Lista Blanca" con la siguiente estructura:
	{
	"idRegistro": "",
	"fechaMensaje": "YYYYMMDDHH24mmss",
	"Resultado": "Registrado",
	"codigoRechazo": ""
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 39:

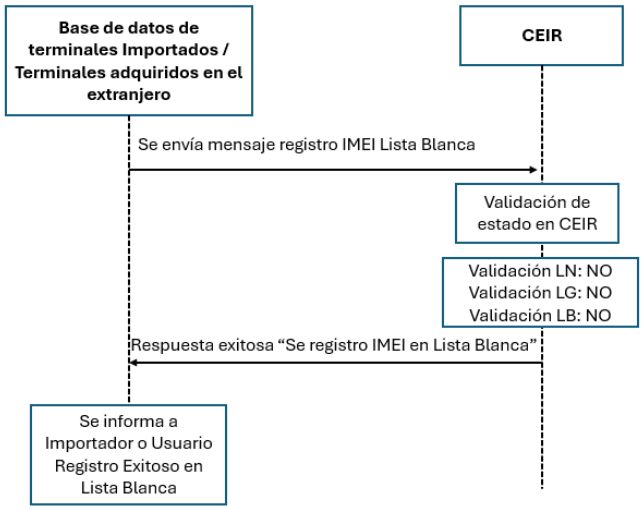


Figura 39: Propuesta de diagrama de estado para el caso de uso B-5
Fuente y elaboración propia

3.2.4.3 Diseño del flujo de intercambio de información para API Sincronización de EIR

Es necesario señalar que se identificó el uso de la API sincronización de EIR en los casos de Uso A-2, A-5 y A-6, por lo que en este punto se mostrará el caso de uso para compartir a todos los concesionarios, mediante mensajes, el resultado del proceso de validación diaria.

Caso de Uso C-1	
Recurso:	API Gestión de acceso a la red
Precondición:	Concesionarios Móviles, remiten archivo con todos los eventos de IMSI attach ocurridos el día anterior.
Acciones:	- Si IMEI está en Lista Negra, se valida que la fecha de vinculación no sea mayor a la fecha de registro en Lista Negra. - Si IMEI está en Lista Gris (Motivo No Registro Lista Blanca), se valida que la fecha de vinculación no sea mayor a 15 días la fecha de registro en Lista Gris, si se cumple esta condición se registra en Lista Negra. - Se valida IMEI exista base de datos GSMA, si no se cumple esta condición se registra en Lista Negra.
Salida:	CeirBrodcast (IMEI-IMSI) considerando la siguiente estructura:

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 40:

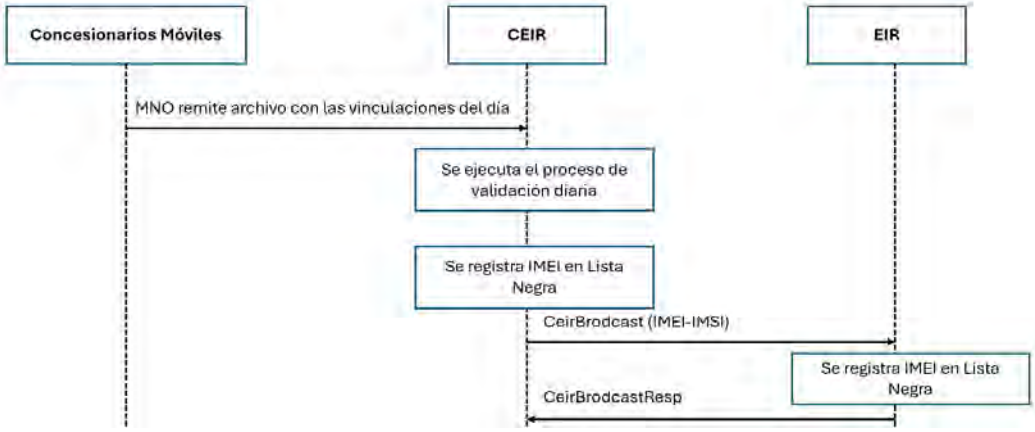


Figura 40: Propuesta de diagrama de estado para el caso de uso C-1
Fuente y elaboración propia

3.2.4.4 Diseño del flujo de intercambio de información para API Gestión de acceso a la red.

En el documento ITU-T Q-Sup.76[7] se definen la estructura de los mensajes denominados LUStatusReq y LUStatusResp, tal como se muestran en las Figuras 41 y 42.

LUStatusReq				
Name	**Type**	**Length**	**Required**	**Description**
MessageHeader	MessageHeaderType		R	Will contain the information of AREA, MNO and Date
LUStatusReq elements				
Triplet			R	This will contain the information of IMEI and IMSI
Triplet/IMEI	Integer	14-16	R	IMEI of the mobile handset being used
Triplet/IMSI	Integer	15	R	IMSI of SIM being used

Figura 41: Definición de la estructura del mensaje LUStatusReq [7]

LUStatusResp				
Name	**Type**	**Length**	**Required**	**Description**
MessageHeader	MessageHeaderType		R	Will contain the information of AREA, MNO and Date
LUStatusResp Elements				
Result	ResultType		R	
FailCause	FailCauseType		C	In case result is failure.
Triplet			R	
Triplet/IMEI	Integer	14-16	R	IMEI of the mobile handset being used
Triplet/IMSI	Integer	15	R	IMSI of SIM being used
Triplet/Status	String	5	C	It can be BLOCKED or PERMITTED

Figura 42: Definición de la estructura del mensaje LUStatusResp [7]

Caso de Uso D-1	
Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI registrado en Lista Negra del CEIR

Entrada:	LUStatusReq(TRIPLETA) considerando la siguiente estructura:
	{
	"concesionario":"", "date":"",
	"imei":"", "imsi":"", "isdn":""
	}

Acciones:	No Aplica

Salida:	LUStatusResp(BLOCKED) considerando la siguiente estructura:
	{
	"date": "YYYYMMDDHH24mmss",
	“Result”: “OK”,
	"codigoRechazo": "",
	"imei":"",
	"imsi":"",
	"status": "BLOCKED"
	}

En este caso, se deniega el acceso a la red, el diagrama de estado para este caso de uso se muestra en la Figura 43.

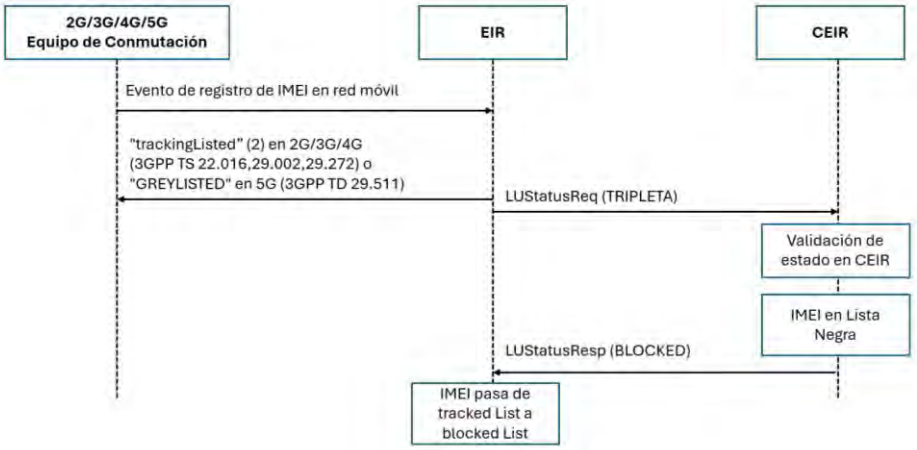


Figura 43: Propuesta de diagrama de estado para el caso de uso D-1
Fuente y elaboración propia

Es necesario señalar que, el siguiente evento que consulte el estado del IMEI al EIR, el EIR denegaría el acceso directamente, tal como se establece en el diagrama de estado de la Figura 44.

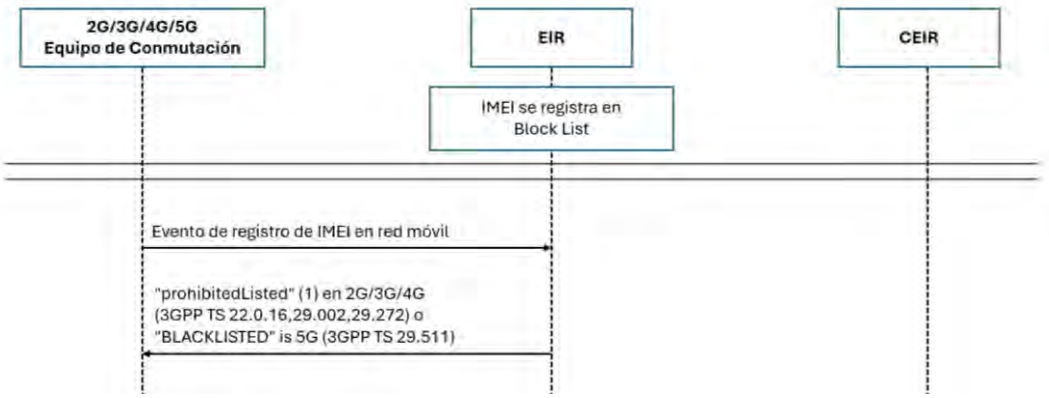


Figura 44: Propuesta de diagrama de estado para el caso de uso D-1
Fuente y elaboración propia

Caso de Uso D-2	
Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI registrado en Lista Gris como Robado

Entrada:	LUStatusReq(TRIPLETA) considerando la siguiente estructura:
	{
	"concesionario":"", "fecha":"",
	"imei":"", "imsi":"", "isdn":""
	}

Acciones:	- Mensaje SMS al Nro servicio “Está haciendo uso de un terminal móvil con IMEI ***, reportado como robado debe entregar este equipo en la comisaría más cercana, y solicitar se entregue la constancia de devolución, en caso no se realice se suspenderá su servicio en 30 días.” Registrar evento en tabla de trazabilidad.

Salida:	LUStatusResp(ALLOWED) considerando la siguiente estructura:
	{
	"date": "YYYYMMDDHH24mmss",
	“Result”: “OK”,
	"codigoRechazo": "",
	"imei":"",
	"imsi":"",
	"status": " ALLOWED "
	}

En este caso, el EIR permite el acceso a la red, sin embargo, todos los eventos de acceso a la red serán remitidos al CEIR y el servicio móvil vinculado recibirá mensajes de advertencia de estar usando un terminal móvil reportado como robado. El primer mensaje tendrá un código OTP de duración 15 días,

para que el usuario cargue a una web la constancia de entrega del terminal a la policía. El diagrama de estado se muestra en la Figura 45.

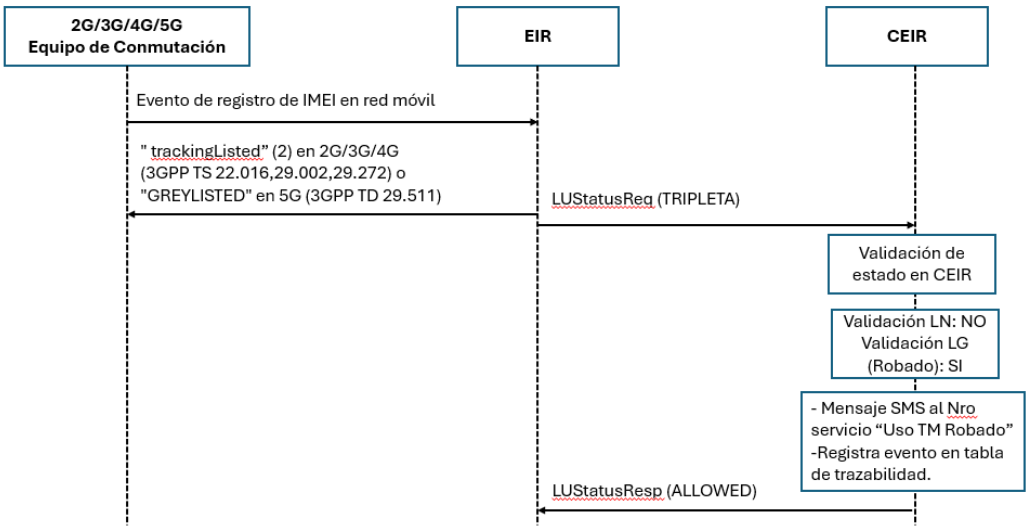


Figura 45: Propuesta de diagrama de estado para el caso de uso D-2
Fuente y elaboración propia

Caso de Uso D-3	
Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI registrado en Lista Gris como No Registrado Lista Blanca

Entrada:	LUStatusReq(TRIPLETA) considerando la siguiente estructura:
	{
	"concesionario":"", "fecha":"",
	"imei":"", "imsi":"", "isdn":""
	}

Acciones:	- Mensaje SMS al Nro servicio Está haciendo uso de un terminal móvil con IMEI **, no registrado en Lista Blanca, si está en proceso de compra, no continúe y devuelva el equipo al vendedor. -Registrar evento en tabla de trazabilidad.
Salida:	LUStatusResp(ALLOWED) considerando la siguiente estructura: {"date": "YYYYMMDDHH24mmss", “Result”: “OK”, "codigoRechazo": "", "imei":"", "imsi":"", "status": " ALLOWED " }

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 46.

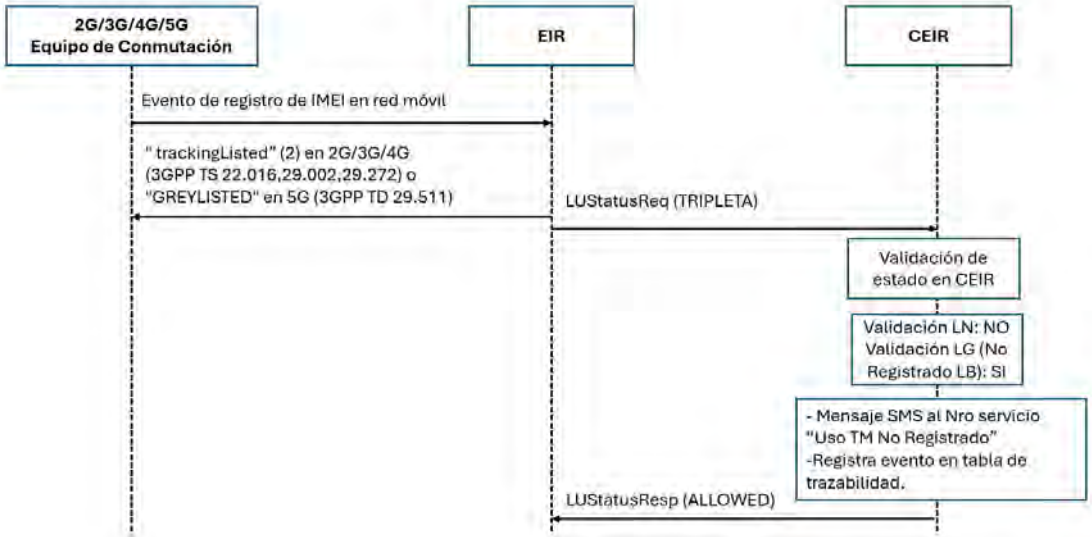


Figura 46: Propuesta de diagrama de estado para el caso de uso D-3
Fuente y elaboración propia

Caso de Uso D-4	
Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI **no** registrado en Lista Gris del CEIR IMEI registrado en Lista Blanca

Entrada:	LUStatusReq(TRIPLETA) considerando la siguiente estructura:
	{
	"concesionario":"", "fecha":"",
	"imei":"", "imsi":"", "isdn":""
	}

Acciones:	No Aplica

Salida:	LUStatusResp (PERMITTED) considerando la siguiente estructura:
	{
	"date": "YYYYMMDDHH24mmss",
	“Result”: “OK”,
	"codigoRechazo": "",
	"imei":"",
	"imsi":"",
	"status": " PERMITTED"
	}

Considerando los detalles del caso de uso, el diagrama de estado se muestra en la Figura 47:

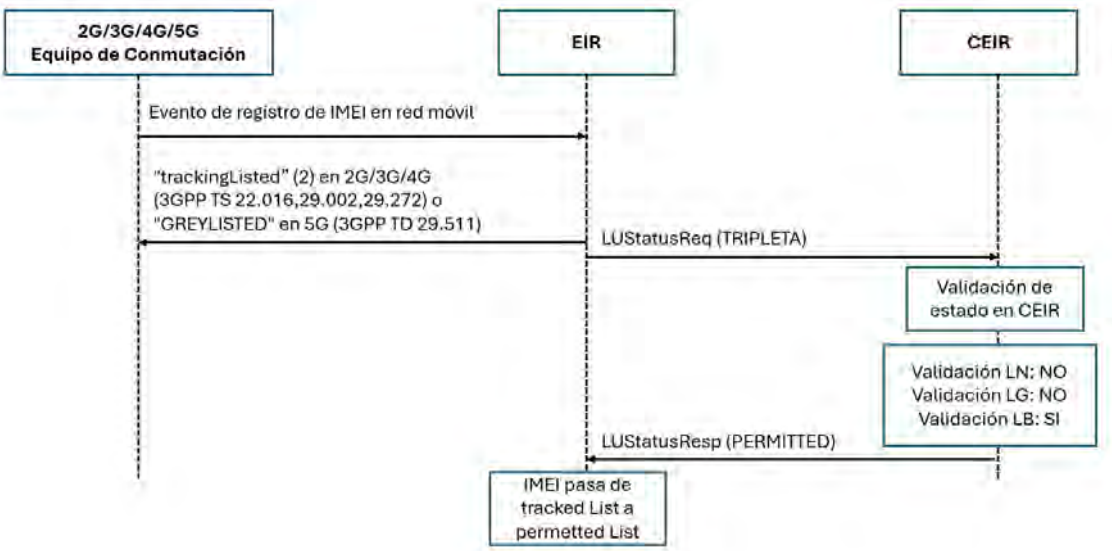


Figura 47: Propuesta de diagrama de estado para el caso de uso 16
Elaboración propia, fuente [6]

Es necesario señalar que, el siguiente evento que consulte el estado del IMEI al EIR, el EIR brindará acceso directamente, ya que en la consulta previa el CEIR notificó el registro del IMEI en la lista blanca del EIR, tal como se observa en el siguiente diagrama de estado.

Figura 48: Continuación del diagrama de estado caso de uso D-4
Fuente y elaboración propia

Caso de Uso D-5	
Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI **no** registrado en Lista Gris del CEIR IMEI **no** registrado en Lista Blanca
Entrada:	LUStatusReq(TRIPLETA) considerando la siguiente estructura: { "concesionario":"", "fecha":"", "imei":"", "imsi":"", "isdn":""}

Acciones:	- Mensaje SMS al Nro servicio “Está haciendo uso del terminal móvil no registrado, en los siguientes días no tendrá acceso a la red” -Registrar evento en tabla de trazabilidad.

Salida:	LUStatusResp (ALLOWED) considerando la siguiente estructura:
	{
	"date": "YYYYMMDDHH24mmss",
	“Result”: “OK”,
	"codigoRechazo": "",
	"imei":"",
	"imsi":"",
	"status": " ALLOWED "
	}

Considerando los detalles de caso de uso, el diagrama de estado se muestra en la Figura 49.

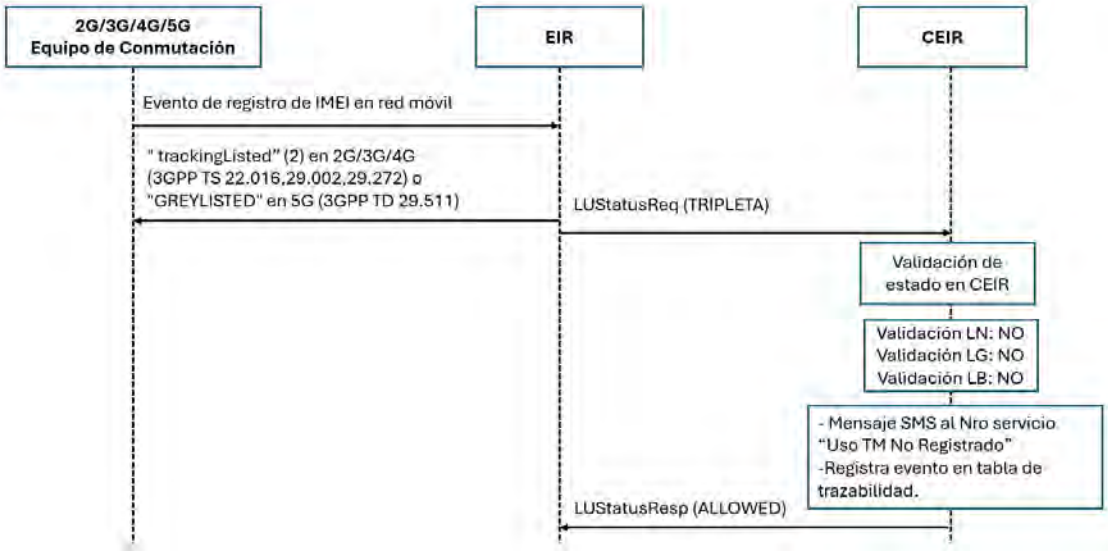


Figura 49: Propuesta de diagrama de estado para el caso de uso D-5
Fuente y elaboración propia

3.3 Validaciones de la propuesta

Considerando la arquitectura propuesta y los casos de uso definidos previamente, se realizará las simulaciones correspondientes a fin de validar los elementos principales de la propuesta. Sin perjuicio de ello, en el Anexo 1, se muestran las pruebas realizadas para cada uno de los casos de uso.

3.3.1 Validación funcional del diseño propuesto

Según lo señalado en los casos de uso, los elementos de entrada en las distintas APIS corresponden a mensajes bajo la estructura JSON. Considerando ello, se usará el software SOAP UI como herramienta para la simulación de envió de mensajes, según la estructura definida en la etapa de diseño.

3.3.1.1 Simulación de intercambio de información de un reporte de robo de un equipo terminal.

Para esta simulación se usa un caso ideal, es decir que el IMEI que se intenta reportar como robado, se encuentre en la lista blanca y no se encuentre en lista negra. Para esto se usó el caso de uso A-2, realizando la simulación de un reporte de bloqueo para el IMEI “777777770000002”

En las figuras 50 y 51, se muestra la consulta a la base de datos a fin de validar el estado del IMEI “777777770000002” en las tablas de Lista Blanca y Lista Negra, respectivamente; según lo establecido en las precondiciones del caso de uso A-2.

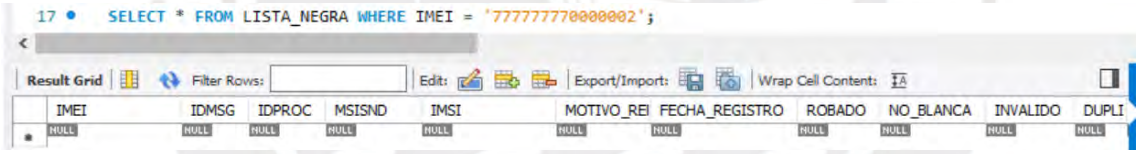


Figura 50: Consulta a Lista Negra, validación de precondición caso de uso A-2

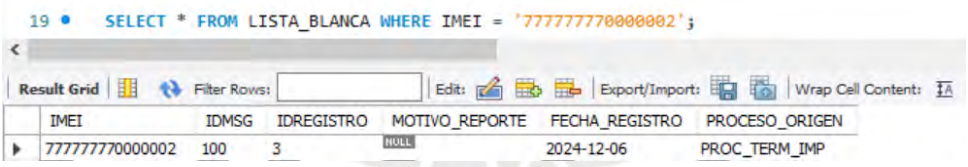


Figura 51: Consulta a Lista Blanca, validación de precondición caso de uso A-2

Luego, se realizó la simulación de envío de mensaje al API “API Bloqueo / Desbloqueo” como se muestra en la Figura 52, validando que el CEIR responde con los argumentos definidos en el caso de uso A-2, es decir se realiza el registro de sustracción.

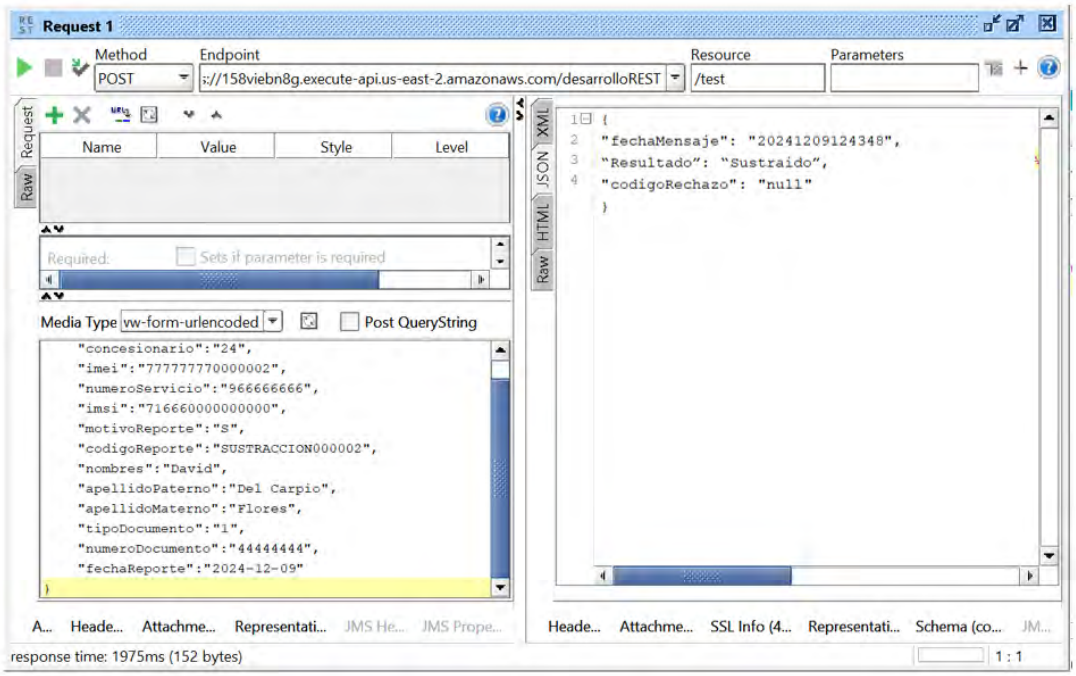


Figura 52: Envío de mensaje de solicitud de bloqueo al CEIR

Como se muestra en la Figura 52, la respuesta es exitosa y el registro del evento quedó almacenado en la tabla MSG_ROBO_RECUP como se muestra en la Figura 53.

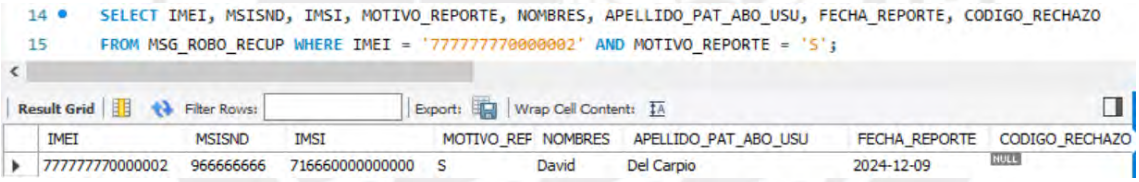


Figura 53: Registro del evento de solicitud de bloqueo en CEIR

Asimismo, se valida que se realiza la acción de eliminación del IMEI en Lista Blanca, como se muestra en la Figura 54, así como, se registra esta eliminación en la tabla Lista Blanca Histórica, como se muestra en la Figura 55.

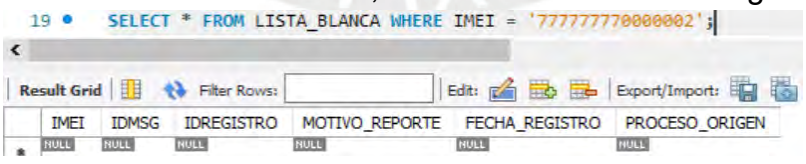


Figura 54: Consulta a tabla Lista Blanca del CEIR, luego de reporte de robo

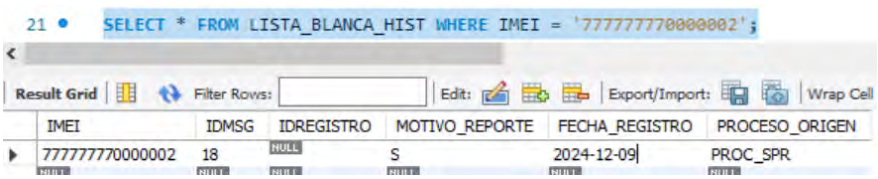


Figura 55: Consulta a tabla Lista Blanca Histórica del CEIR, luego de reporte de robo

Para completar esta simulación, se consulta la tabla Lista Gris del CEIR y se valida que el IMEI remitido en el reporte de robo se encuentra registrado como se muestra en la Figura 56.

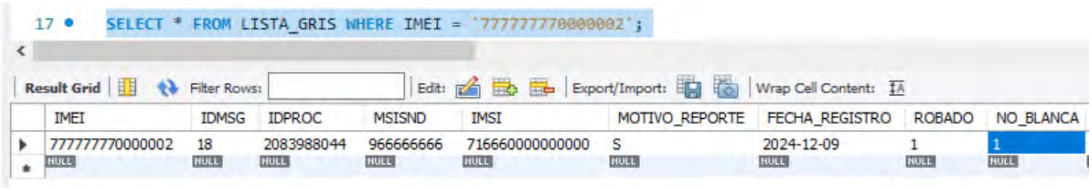


Figura 56: Consulta a tabla Lista Gris del CEIR, luego de reporte de robo

Al completar la simulación de intercambio de información de un reporte de robo de un equipo terminal, se valida la correcta funcionalidad según los requerimientos definidos en la propuesta del proceso de reporte de robo de equipos terminales.

3.3.1.2 Simulación de solicitud a CEIR de un acceso permitido a la red móvil.

Para la simulación se consideró el detalle establecido en el caso de uso D-4, el cual considera como respuesta del API de Gestión de acceso a la red el mensaje “PERMITTED”. En ese sentido, se usó el IMEI “777777770000001”.

En las figuras 57 y 58, se muestra la consulta a la base de datos a fin de validar el estado del IMEI “777777770000001” en las tablas de Lista Negra y Lista Gris, respectivamente, según lo establecido en las precondiciones del caso de uso D-4, es decir, que el referido IMEI no se encuentre registrado en dichas listas.

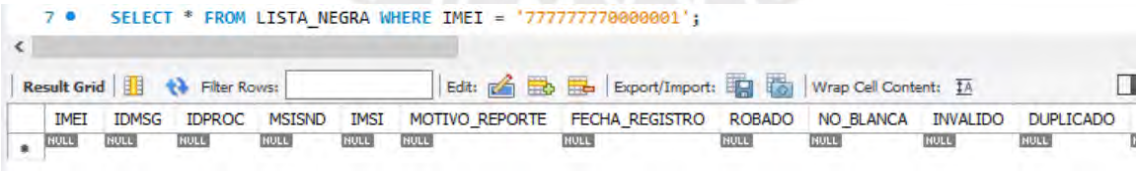


Figura 57: Consulta a Lista Negra, validación de precondición caso de uso D-4

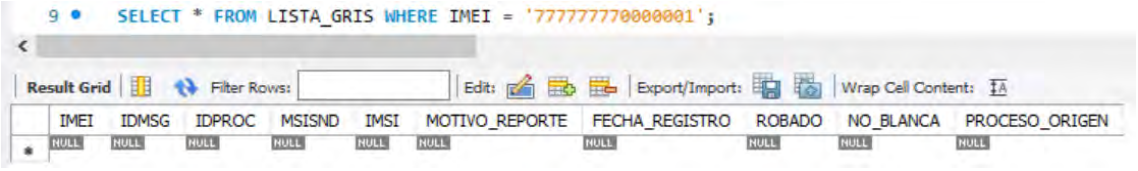


Figura 58: Consulta a Lista Gris, validación de precondición caso de uso D-4

En la figura 59, se muestra la consulta a la base de datos a fin de validar que IMEI “777777770000001” se encuentre registrado en Lista Blanca, según lo establecido en la precondición del caso de uso D-4.

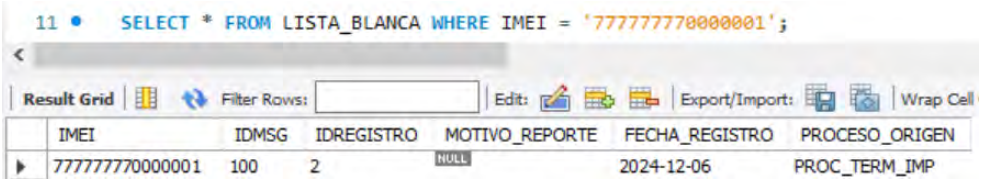


Figura 59: Consulta a Lista Blanca, validación de precondición caso de uso D-4

Luego, se realizó la simulación de envío de mensaje al API “Gestión de acceso a la red”, como se muestra en la Figura 59, validando que el CEIR responde con los argumentos definidos en el caso de uso D-4, brindando la respuesta “PERMITTED”

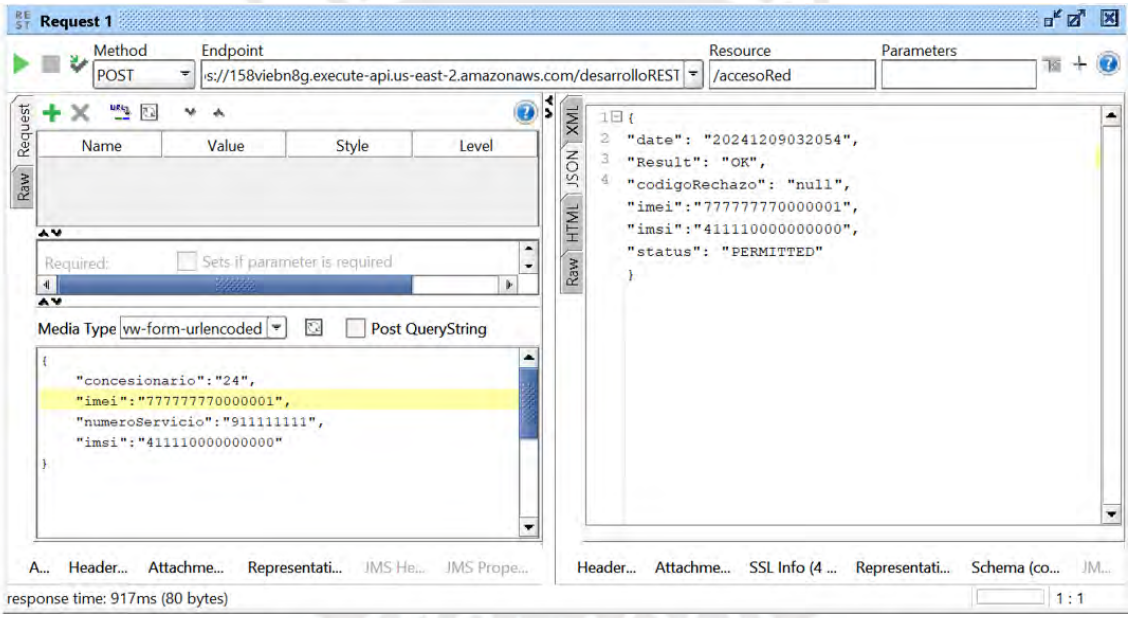


Figura 60: Envío de mensaje acceso permitido a la red móvil.

Al finalizar la simulación de intercambio de información de una solicitud a CEIR de un acceso permitido a la red móvil, se valida la correcta funcionalidad, según los requerimientos definidos en el diseño del proceso de gestión de acceso a la red móvil.

3.3.1.3 Simulación de intercambio de información para un acceso denegado a la red móvil.

Para la simulación se consideró el detalle establecido en el caso de uso D-1 que considera como respuesta del API de

Gestión de acceso a la red, el mensaje “BLOCKED”. En ese sentido, se usó el IMEI “987654321987654”.

En la figura 61, se muestra la consulta a la base de datos, a fin de validar el estado del IMEI en la tabla de Lista Negra, según lo establecido en las precondiciones del caso de uso D-1. Es decir que, el referido IMEI se encuentre registrado en lista negra.

IMEI	MSISND	IMSI	MOTIVO_REPORTE	FECHA_REGISTRO	ROBADO	NO_BLANCA	INVALIDO	DUPLI
987654320000000	999999999	789450000000000	P	2024-12-06	1	0	0	0
987654321987654	999999999	789456123789456	S	2024-12-06	1	1	0	0

Figura 61: Consulta a Lista Negra, validación de precondición caso de uso D-1

Luego, se realizó la simulación de envío de mensaje al API “Gestión de acceso a la red”, como se muestra en la Figura 62, validando que el CEIR responde con los argumentos definidos en el caso de uso D-1.

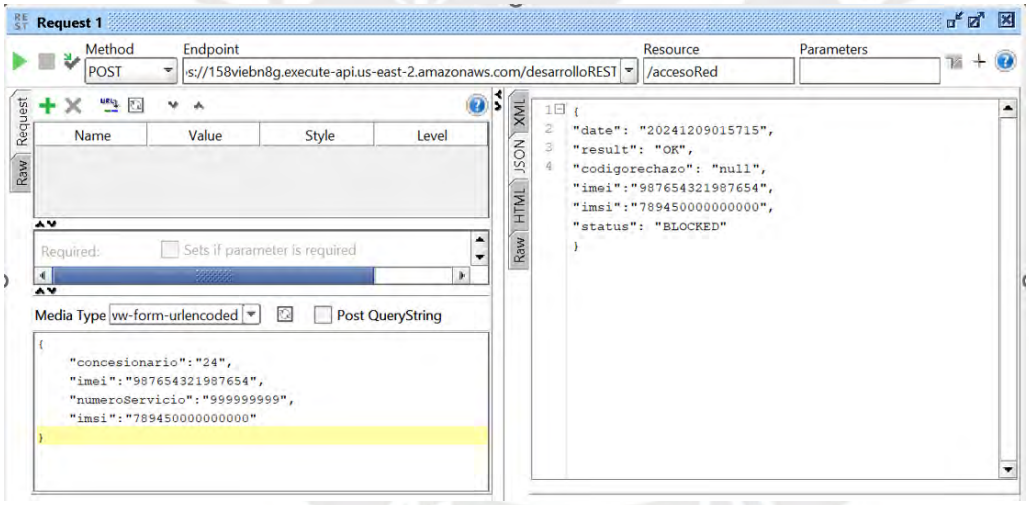


Figura 62: Envío de mensaje acceso denegado a la red móvil.

Al completar la simulación de intercambio de información de una solicitud a CEIR de un acceso denegado a la red móvil, se valida la correcta funcionalidad, según los requerimientos definidos en el diseño del proceso de gestión de acceso a la red móvil.

3.3.1.4 Simulación de información para la trazabilidad de terminales móviles para su recuperación.

Para la simulación, se consideró el detalle establecido en el caso de uso D-2 que considera como respuesta del API

de Gestión de acceso a la red, el mensaje “ALLOWED”. En ese sentido, se usó el IMEI “777777770000008”.

En las figuras 63 y 64, se muestra la consulta a la base de datos a fin de validar el estado del IMEI en la tabla de Lista Gris y Lista Negra, respectivamente, según lo establecido en las precondiciones del caso de uso D-2.

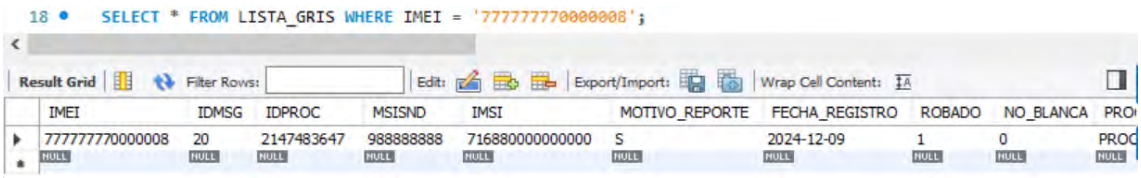


Figura 63: Consulta a Lista Gris, validación de precondición caso de uso D-2

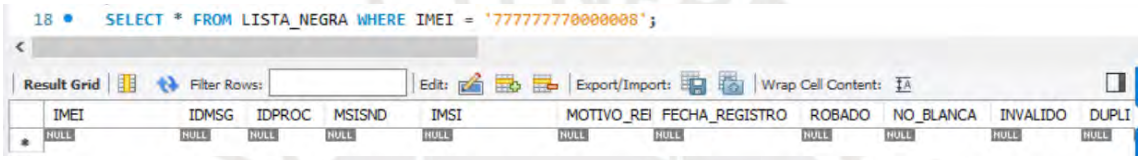


Figura 64: Consulta a Lista Negra, validación de precondición caso de uso D-2

Luego, se realizó la simulación de envío de mensaje al API “Gestión de acceso a la red”, como se muestra en la Figura 65, validando que el CEIR responde con los argumentos definidos en el caso de uso D-2.

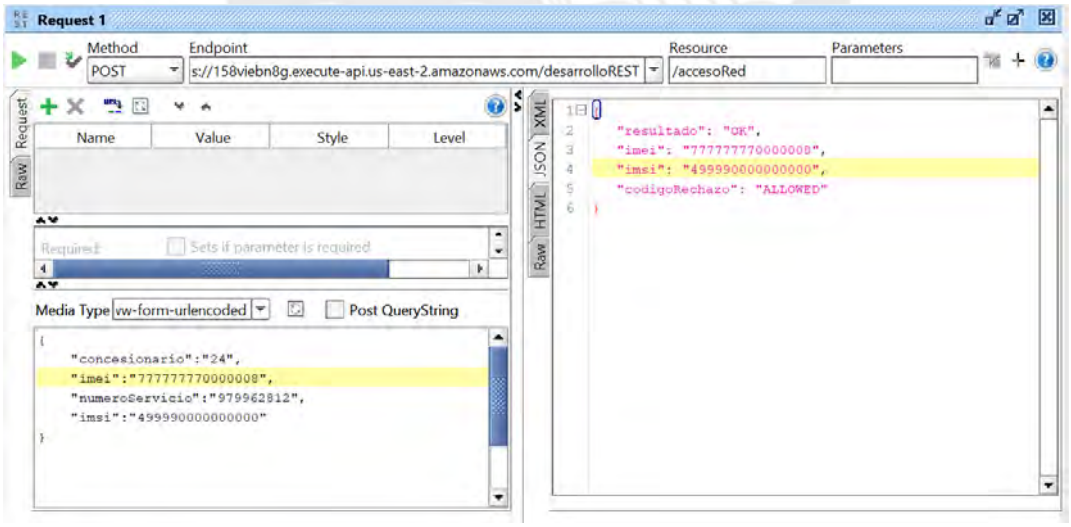


Figura 65: Envío de mensaje acceso temporal a la red móvil, registro en trazabilidad

Al completar la simulación de intercambio de información de una solicitud a CEIR de un acceso temporal a la red móvil, se valida la correcta funcionalidad, según los requerimientos definidos en el diseño del proceso de

gestión de acceso a la red móvil, observando que todos los eventos de IMEI registrados en Lista Gris son almacenados en la tabla de Consulta CEIR, como se muestra en la Figura 66. Adicionalmente, el envío de los SMS correspondiente se muestra en la Figura 67.

```
4 •  SELECT * FROM CONSULTAS_CEIR WHERE IMEI = '777777770000008';
5
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ID_CONSULTA	CONCESIONARIO	IMEI	MSISND	IMSI	FECHA_REGISTRO	COD_CODIGO_RECHAZO
7	24	777777770000008	979962812	499990000000000	2024-12-13	NULL
9	24	777777770000008	979962812	499990000000000	2024-12-13	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 66 – Registros en la Tabla Consultas_CEIR IMEI en Lista Gris

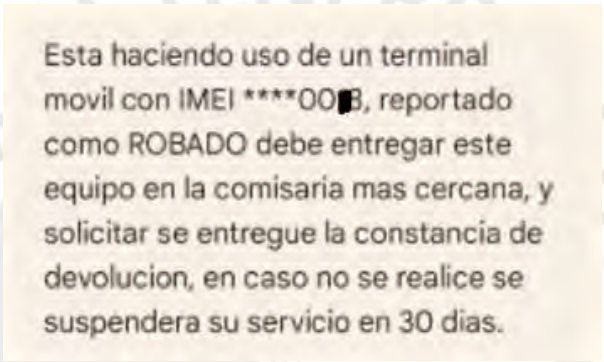


Figura 67 – SMS de advertencia uso de terminal móvil robado

3.3.2 Evaluación de la eficiencia del diseño propuesto para combatir el robo de terminales móviles

Como se señala en el numeral 3.1.4 del presente documento, actualmente no existe un enfoque multisectorial que considere procesos que permitan incrementar las recuperaciones. Considerando ello, en la Tabla 11 se detalla las diferencias a nivel de funcionalidad entre el sistema actual y el sistema propuesto.

Funcionalidad	Solución Actual	Solución Propuesta
Reporte de robos de terminales móviles	El sistema recibe información en tiempo real, sin embargo, el IMEI se registra inmediatamente a la Lista Negra, descartando la posibilidad de poder detectar eventos en la red móvil posteriores al reporte de robo.	El sistema recibe información en tiempo real. Se informa a los concesionarios móviles el acceso temporal a la red móvil (Lista Gris), de manera que se puedan registrar eventos en la red móvil posteriores al reporte de robo, que permitan identificar a la persona

Funcionalidad	Solución Actual	Solución Propuesta
		que podría tener el equipo robado.
Información a usuarios de Terminales móviles no registrados en Lista Blanca	El sistema actual realiza una validación diaria, entre otras funcionalidades es posible detectar si existe equipos no registrados en Lista Blanca. Los usuarios recibirán algunos días después de empezar a usar un nuevo equipo, un SMS que informa que el equipo se bloquearía por no estar registrado en Lista Blanca.	Siendo que se contaría con un proceso de gestión de acceso a la red en tiempo real, los usuarios son alertados de que el terminal móvil no esta registrado en Lista Blanca, de inmediato, luego de que el servicio móvil se vincula al terminal.
Reporte de robos de terminales móviles que tengan ESIM	El sistema no hace diferencias, aplica el mismo procedimiento para todo tipo de terminal móvil.	Al registrar eventos en la red móvil posteriores al robo y que el dispositivo no puede ser apagado hasta que se pueda retirar la batería, es posible que las autoridades Judiciales apliquen el proceso de geolocalización y tener información de lugares donde se llevan los equipos robados.

Así, para determinar el valor cuantitativo de la eficiencia del sistema propuesto, según la experiencia comparada en otros países, sería necesario realizar un piloto de la implementación de la Lista Gris y de los procesos de trazabilidad, considerando que en Brasil este piloto generó un porcentaje de recuperación mayor al 20% [22] y en algunas regiones de India seria de hasta el 50% [13].

3.4 Propuesta de modificación del Decreto Legislativo N° 1338

La modificación normativa contempla la habilitación de la Lista Gris para la trazabilidad de terminales móviles robados y la inclusión de la SUNAT como actor principal en el proceso de registro de terminales importados. Considerando ello, se plantea el ajuste normativo de los artículos 4, 6 y 8 del Decreto Legislativo N° 1338 y la incorporación de disposiciones complementarias finales a dicho decreto según se señala a continuación:

Artículo 4. Contenido del RENTESEG

(…)

4.4 Se incluyen en la Lista Gris los terminales móviles reportados como perdidos o sustraídos o no registrados en Lista Blanca; los cuales se habilitan para operar en la red móvil por un periodo establecido, conforme a la Metodología aprobada por el Ministerio del Interior, con la finalidad de realizar la trazabilidad y recuperación del equipo terminal. La reincorporación de los terminales móviles a la Lista Blanca corresponde adicionalmente la eliminación en la Lista Gris.

El Ministerio Público podrá requerir la geolocalización y CDR de los terminales móviles que se encuentren en Lista Gris como sustraídos siempre que se detecte un evento de acceso a la red móvil posterior al reporte de sustracción.

4.5 Se incluyen a la Lista Negra los terminales móviles reportados como perdidos, sustraídos **que no se recuperen como máximo en 100 días desde que se generó el reporte de sustracción. En caso se detecten eventos de acceso a la red móvil posteriores al reporte de sustracción, el concesionario móvil remitirá un SMS informativo al abonado, respecto a la suspensión del servicio móvil vinculado. A los 30 días de la remisión del SMS se realiza la suspensión del servicio**. La reincorporación de los equipos terminales móviles a la Lista Blanca corresponde adicionalmente la eliminación en la Lista Negra.

(…)

Artículo 6. Autoridades competentes

(…) **6.4 Son atribuciones de la SUNAT**

a) Registrar la información de los identificadores únicos de cada terminal importados al momento que ingresen al Perú y de su importador.

b) Registrar la información de terminales móviles adquiridos en el extranjero para uso personal, conforme al procedimiento aprobado por el Ministerio del Interior en el Reglamento del presente Decreto Legislativo.

(…)

Artículo 8. Empresas operadoras de servicios públicos móviles de telecomunicaciones

(…) c) Suspender el servicio móvil asociado al terminal móvil reportado como perdido, sustraído o inoperativo, **o cuando el sistema RENTESEG detecte que el servicio móvil se ha vinculado a un equipo terminal registrado en la Lista Gris como sustraído.** (…)

Disposiciones Complementarias Finales

Primera.-
El presente Decreto Legislativo entrará en vigencia a los 90 días hábiles de publicada la modificación del Reglamento de este Decreto. Luego de dicho plazo, la lista gris ingresará en un periodo de marcha blanca por un periodo de 90 días hábiles adicionales.

Segunda.-
En un plazo de 30 días hábiles, el Ministerio del Interior publica la modificación del Reglamento del Decreto Legislativo N°1338, el cual incluirá la metodología para la determinación del periodo de trazabilidad de los equipos terminales sustraídos o perdidos que hayan ingresado a la lista gris.

Conforme se aprecia, la referida modificación normativa contemplará un periodo de adecuación de 90 días hábiles posteriores a la publicación de la modificación del Reglamento del Decreto Legislativo N°1338, así como un periodo de marcha blanca de 90 días hábiles contados desde la entrada en vigencia del Decreto Legislativo, de manera que se pueda medir la eficiencia del sistema para incrementar la cantidad de recuperaciones. Dicho piloto también serviría para determinar el procedimiento que deberían seguir la Policía Nacional, el Ministerio Publico y las empresas operadoras para una mayor efectividad en el proceso de trazabilidad.

CONCLUSIONES

1. **A nivel de análisis de la problemática**, se identifica como problema central el limitado nivel de recuperación de terminales móviles robados en el Perú, luego se desarrollan alternativas que se enfocan en siguientes medios fundamentales:

    - Presencia de detección en la red móvil de terminales robados para su recuperación.
    - Amplia capacidad para identificar eventos efectuados en un terminal robado para su recuperación.
    - Participación de la SUNAT en el marco normativo vigente, como un actor principal en el proceso de ingreso de terminales móviles a Perú.

2. **Como parte del análisis comparado internacional,** se identifica que los pilotos de implementación de la Lista Gris incrementaron la eficiencia de la recuperación de terminales móviles robados en un 20% para el caso de Brasil y hasta un 50% para el caso de India, lo cual se contrasta con el 7% de recuperaciones que se realizan con la solución tecnológica actual de Perú.

3. **Como resultado del análisis de la situación actual**, se identifica las estrategias para mejorar las oportunidades y mitigar las debilidades de la solución tecnológica actual.

    - La primera estrategia propuesta es la inclusión de la Lista Gris en el proceso de gestión de acceso a la red móvil, el cual se propone realizar en tiempo real (modelo CEIR de API sincrónico).

    - La segunda estrategia es la implementación del modelo de registro restrictivo por parte de la SUNAT en el proceso de registro de terminales importados y adquiridos en el extranjero.

4. **A nivel de diseño de la solución propuesta**, se define los procesos de la solución tecnológica incluyendo el proceso de gestión de acceso a la red móvil, validando que, habilitando la Lista Gris, es posible registrar los eventos en la red móvil posteriores al reporte de sustracción, lo cual permitirá a las autoridades del Ministerio del Interior y Policía Nacional requerir los CDR y geolocalización que permita identificar a las personas vinculadas al terminal móvil robado.

5. **Asimismo, en el diseño de la solución**, se define el modelo de la base de datos relacional del CEIR que gestionará la información de la solución tecnológica referida a los procesos: i) Gestión de acceso a la red móvil para

la trazabilidad, ii) Registro de terminales que ingresan a Perú, iii) Registro de reportes de robo, y iv) sincronización de EIR, validando que el modelo cumple con los requisitos de integridad, y tiempo de respuesta (menor a 30 segundos para la operación en tiempo real del CEIR).

6. **Como resultado del diseño de la solución**, respecto de las aplicaciones para el intercambio de información entre el CEIR y los componentes externos que interactúan con la solución tecnológica, se definieron los requerimientos funcionales de los siguientes servicios:

   a) API Gestión de acceso a la red, permitirá intercambiar información entre los EIR y el CEIR, cada vez que se genere un evento IMSI Attach o Location Update en la red móvil.
   b) API Bloqueo / Desbloqueo, permitirá que los concesionarios móviles reporten al CEIR el registro a Lista Gris por robo o perdida de un equipo terminal, así como el retiro de dicha lista por recuperación.
   c) API Registro Lista Blanca, permitirá el registro en la lista blanca del CEIR una vez que se complete el proceso de registro de terminales importados o el registro de terminales adquiridos en el extranjero por personas naturales.
   d) API Sincronización de EIR, permitirá intercambiar información entre el CEIR y los EIR cuando el CEIR realice una acción de inserción o retiro en la Lista Negra.

7. **A nivel de resultados del diseño de la solución tecnológica**, se valida que el sistema propuesto cumple los requerimientos funcionales establecidos. Dichas validaciones se realizaron mediante la simulación de i) la generación de eventos de acceso a la red móvil, evidenciado el registro de información para la trazabilidad de terminales previamente reportados como robados, así como ii) la generación de eventos para el reporte de robo de un terminal móvil, evidenciando el registro de un terminal en la lista gris del sistema.

8. Se considera que la solución tecnológica propuesta cumple los requisitos funcionales para -potencialmente- mejorar de forma significativa la eficiencia de solución actual de Perú. Por ello, se proponen, además, modificaciones al marco normativo correspondiente y se busca concientizar a las autoridades respecto de este problema público altamente pernicioso.

RECOMENDACIONES

Se recomienda a las entidades que lideran las políticas públicas relacionadas a combatir el robo de terminales móviles, realizar los ajustes en el marco normativo, para incluir a la SUNAT como ente responsable en la declaración de terminales adquiridos en el extranjero para uso personal, de manera que se pueda combatir el comercio ilegal de terminales a través de la implementación de soluciones digitales.

Se recomienda realizar un piloto de implementación de la Lista Gris en Perú, en el marco de la modificación normativa al Decreto Legislativo N° 1338, propuesto en el presente trabajo; de manera que se permita la creación de la Lista Gris en el sistema RENTESEG y la habilitación en los EIR de los Concesionarios Móviles.

Para la operación de dicho piloto, el sistema RENTESEG informará mediante su actual API de bloqueo por sustracción o perdida, si un IMEI debe registrarse en la Lista Negra o Lista Gris de los Concesionarios móviles. El Ministerio Publico y el OSIPTEL requerirán la información de la trazabilidad generada en la lista Gris y los CDR asociados, a fin de identificar a los servicios móviles que generaron vinculaciones luego del reporte de sustracción, identificando a las personas asociadas al hecho delictivo. Asimismo, el Ministerio Público podrá requerir la geolocalización de los servicios móviles identificados en el proceso de trazabilidad.

Finalmente, luego de validar si la eficiencia en la recuperación de terminales móviles del piloto implementado es superior a la eficiencia de la solución actual, considerando que la experiencia en otros países llego a 50%, las entidades que lideran las políticas públicas relacionadas a combatir el robo de terminales móviles en el Perú deben considerar la implementación del CEIR de API sincrónico cuyo diseño se detalla en el presente trabajo.

REFERENCIAS

[1] N.Mandela, T.Wangchuk, T.Mbinda, K.Damedjate, G.Mwendwa, J.Makopa "IMEI-based Mobile Device Tracking and Stolen Phone Identification System“, *11th International Conference on Computing for Sustainable Global Development (INDIACom),* pp. 1115-1121 Feb, 2024.

[2] J.Venegas, V.Zambrano, M.Alonso, N.Guerra, M.Caballero, H.Ortiz "Estudio para identificar acciones para disminuir el robo de equipos terminales móviles" Instituto Federal de Telecomunicaciones, Ciudad de México, México, 2023.

[3] Comisión de regulación de comunicaciones Republica de Colombia, Noviembre 2021, Simplificación del Marco Regulatorio para la restricción de equipos terminales hurtados [Online] Available: https://www.crcom.gov.co/system/files/Proyectos%20Comentarios/2000-71-17/Propuestas/documento_soporte_simplificacion_del_marco_regulatorio_para_la_restriccion_de_equipos_terminales_hurtados.pdf

[4] Unión Internacional de Telecomunicaciones ITU, Setiembre 2020, ITU-T Q.5052 Addressing mobile devices with a duplicate unique identifier.

[5] Unión Internacional de Telecomunicaciones ITU, Diciembre 2021, ITU-T Q-Sup.75 Use cases on the combat of counterfeit ICT and stolen mobile devices.

[6] Federal Communications Commission FCC, Diciembre 2015, Report of Technological Advisory Council (TAC) Subcommittee on Mobile Device Theft Prevention (MDTP) Analysis and Recommendations for 2015.

[7] Unión Internacional de Telecomunicaciones ITU, Octubre 2023, ITU-T Q-Sup.76 Common approaches and interfaces for data exchange between the central equipment identity register and the equipment identity register.

[8] Unión Internacional de Telecomunicaciones ITU, Diciembre 2021, ITU-T Q-Sup.75 Use cases on the combat of counterfeit ICT and stolen mobile devices.

[9] ANATEL (Julio 2024) Nova coleta, os dados de acceso por municipio [Online] Available: https://informacoes.anatel.gov.br/paineis/acessos/telefonia-movel

[10] ANATEL (Julio 2024) Celular Seguro alcança marca de 60 mil bloqueios após alertas de usuarios [Online] Available: https://www.gov.br/mj/pt-br/assuntos/noticias/celular-seguro-alcanca-marca-de-60-mil-bloqueios-apos-alertas-de-usuarios

[11] Departamento Administrativo Nacional de Estadística de Colombia (Marzo 2023) Encuesta de convivencia y seguridad ciudadana [Online] Available: https://www.dane.gov.co/files/investigaciones/poblacion/convivencia/2021/Presentacion_ECSC_2021.pdf

[12] Department of Telecommunications – India (Julio 2024) How to block a lost/stolen phone [Online] Available: https://www.ceir.gov.in/Home/index.jsp

[13] Department of Telecommunications – India (Agosto 2024), State leads nation in mobile recovery rates [Online] Available: https://indianexpress.com/article/cities/hyderabad/ceir-portal-telangana-leads-recoveries-lost-stolen-mobile-phones-8882783/

[14] Government of India Ministry of Communications (Julio 2024), Telephone Subscribers – Wireless Subscribers [Online] Available: https://dot.dashboard.nic.in/DashboardF.aspx

[15] Organismo Supervisor de la Inversión Privada en Telecomunicaciones, Abril 2024, Instructivo técnico para la implementación del RENTESEG.

[16] V.M López Fandiño, Sistemas de Big Data, Madrid:RA-MA Editorial, 2023

[17] J. Chapin, M. Roberts,Programar AWS Lambda, Sebastopol:O'Reilly Media,2020

[18] K. C. Wang, Embedded and Real-Time Operating Systems, Washington: Springer, 2023

[19] K.Siva Prasad, Beginning Spring Boot 2, New York: Apress, 2017.

[20] T. Hunter, Advanced Microservices: A Hands-on Approach to Microservices Infrastrusture and Tooling, New York: Apress, 2017

[21] P. Belohlavek. OEE: Overall Equipment Effectiveness. Su abordaje unicista, Buenos Aires: Blue Eagle Group, 2006

[22] J. A. Zanon, comunicación privada, Oct 2024.

[23] Experto identidad reservada, comunicación privada, Marzo 2025.

[24] Organismo Supervisor de la Inversión Privada en Telecomunicaciones, Octubre 2024, PUNKU OSIPTEL [Online]: Available https://punku.osiptel.gob.pe/

ANEXOS

ANEXO 1: Pruebas de funcionalidad de los casos de uso

Caso de Uso A-1

Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado registrado en Lista Negra.

Validación, el IMEI se encuentre en lista negra

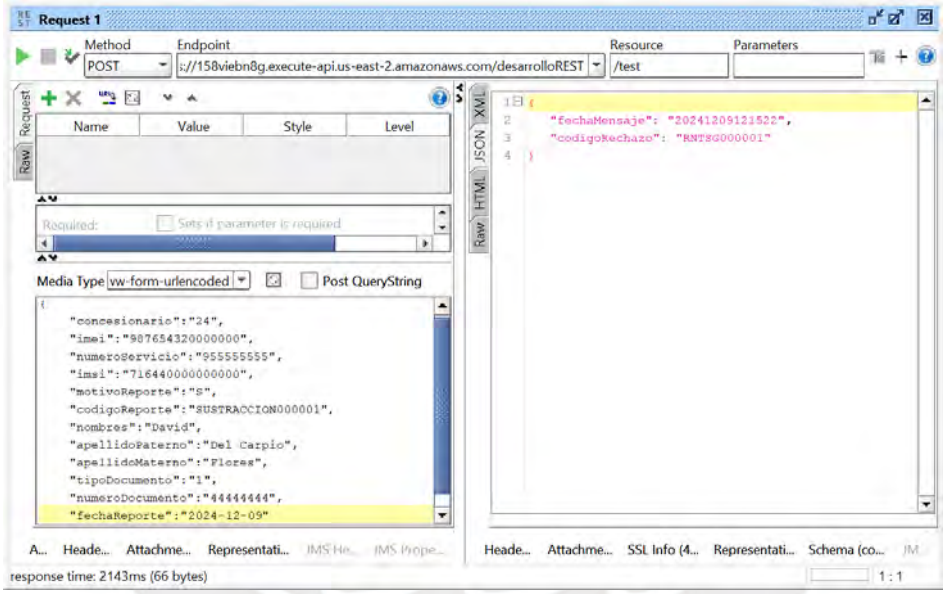
```
7 •  SELECT * FROM LISTA_NEGRA WHERE IMEI = '987654320000000';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	MSISND	IMSI	MOTIVO_REI	FECHA_REGISTRO	ROBADO	NO_BLANCA	INVALIDO	DUPLI	PROCESO_ORIGEN
987654320000000	999999999	789450000000000	S	2024-12-06	1	0	0	0	PROC_VAL_DIARIA
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Entrada:	Reporte de Robo

Se realiza la prueba en el API “API Bloqueo / Desbloqueo”, validando que el CEIR responde con los argumentos definidos en el caso de uso A-1.



El intento de bloqueo quedo registrado en la tabla MSG_ROBO_RECUP

```
14 •  SELECT IMEI, MSISND, IMSI, MOTIVO_REPORTE, NOMBRES, APELLIDO_PAT_ABO_USU, FECHA_REPORTE, CODIGO_RECHAZO
15    FROM MSG_ROBO_RECUP WHERE IMEI = '987654320000000' AND MOTIVO_REPORTE = 'S';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

IMEI	MSISND	IMSI	MOTIVO_REP	NOMBRES	APELLIDO_PAT_ABO_USU	FECHA_REPORTE	CODIGO_RECH
987654320000000	955555555	716440000000000	S	David	Del Carpio	2024-12-09	RNTSG000001

En tabla GEN_CODIGO_RECHAZO se muestra la descripción del código de rechazo generado por el sistema

```
5 •  SELECT * FROM GEN_CODIGO_RECHAZO WHERE CODIGO = 'RNTSG000001';
6
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wra

CODIGO	DESCRIPCION
RNTSG000001	Bloqueo sobre bloqueo, IMEI ya está en LG o LN.
NULL	NULL

Acciones:	No aplica

Caso de Uso A-2

Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado **no** registrado en Lista Negra. IMEI reportado registrado en Lista Blanca.

Validación, el IMEI no se encuentre en lista negra

```
17 •  SELECT * FROM LISTA_NEGRA WHERE IMEI = '777777770000002';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDPROC	MSISND	IMSI	MOTIVO_REI	FECHA_REGISTRO	ROBADO	NO_BLANCA	INVALIDO	DU
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Validación, el IMEI se encuentra en lista blanca

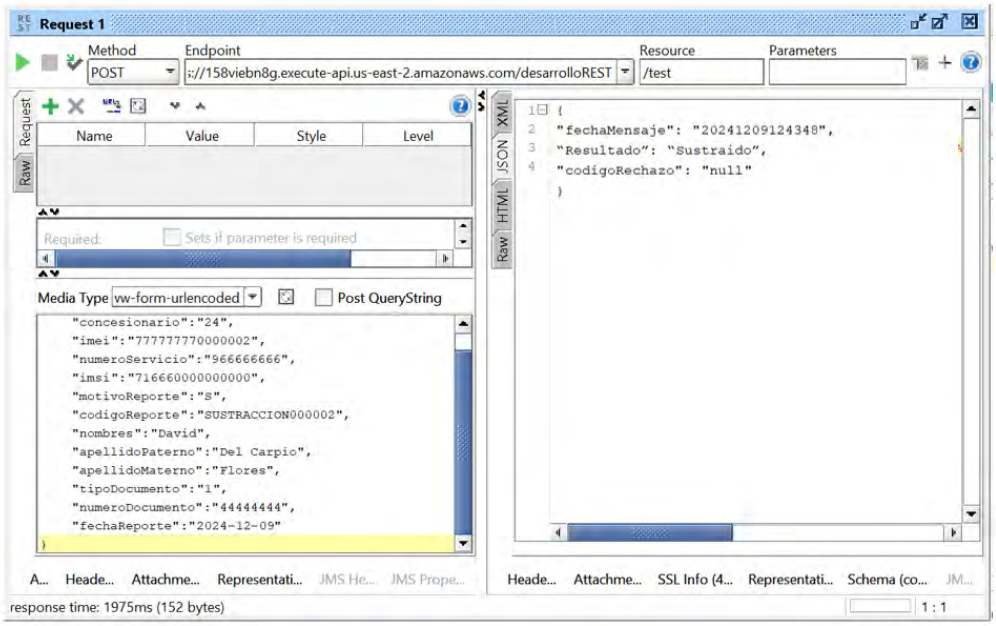
```
19 •  SELECT * FROM LISTA_BLANCA WHERE IMEI = '777777770000002';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDREGISTRO	MOTIVO_REPORTE	FECHA_REGISTRO	PROCESO_ORIGEN
777777770000002	100	3	NULL	2024-12-06	PROC_TERM_IMP

Entrada:	Reporte de Robo

Se realiza la prueba en el API “API Bloqueo / Desbloqueo”, validando que el CEIR responde con los argumentos definidos en el caso de uso A-2.



Se genera una respuesta exitosa, este registro quedo registrado en la tabla MSG_ROBO_RECUP

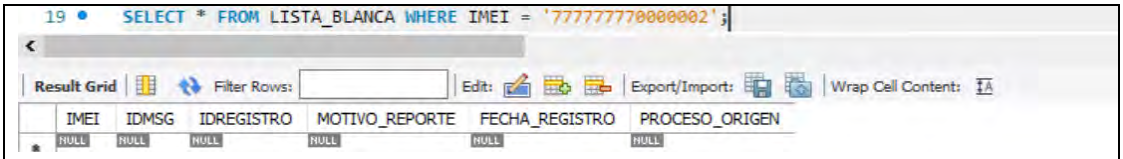
```
14 •  SELECT IMEI, MSISND, IMSI, MOTIVO_REPORTE, NOMBRES, APELLIDO_PAT_ABO_USU, FECHA_REPORTE, CODIGO_RECHAZO
15    FROM MSG_ROBO_RECUP WHERE IMEI = '777777770000002' AND MOTIVO_REPORTE = 'S';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

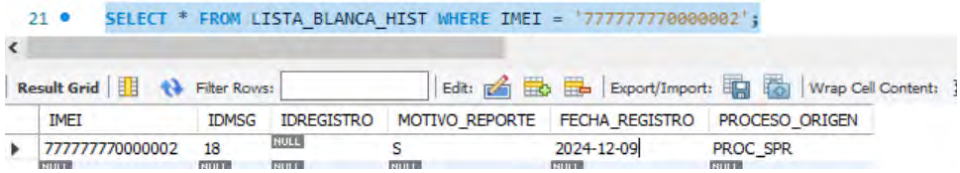
IMEI	MSISND	IMSI	MOTIVO_REP	NOMBRES	APELLIDO_PAT_ABO_USU	FECHA_REPORTE	CODIGO_RECH
777777770000002	966666666	716660000000000	S	David	Del Carpio	2024-12-09	NULL

Acciones:	- Eliminar IMEI de lista blanca. - Guardar el registro de eliminación en la tabla Lista_Blanca_Hist. - Registrar el IMEI en Lista_Gris.

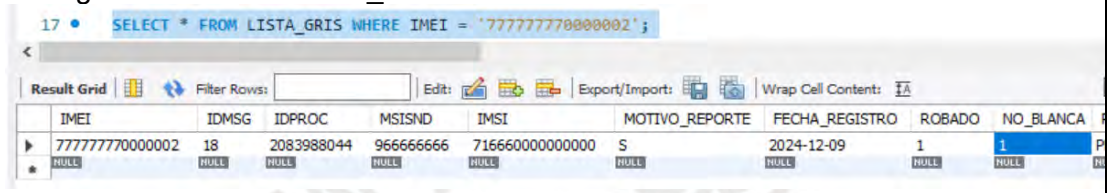
Se elimina el IMEI de lista blanca



Se guarda el registro de eliminación en la tabla Lista_Blanca_Hist.



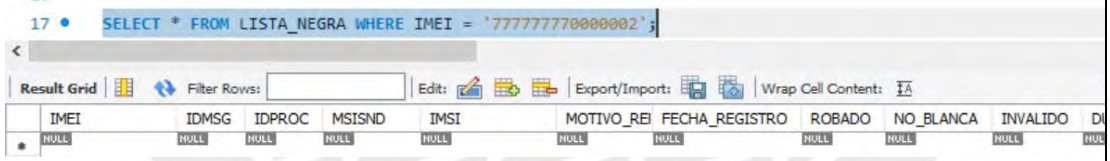
Se registrar el IMEI en Lista_Gris.



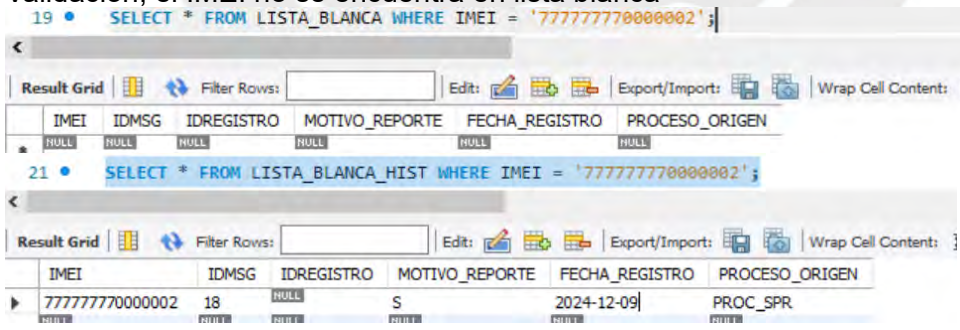
Caso de Uso A-3

Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado **no** registrado en Lista Negra. IMEI reportado **no** registrado en Lista Blanca. IMEI reportado registrado en Lista Gris

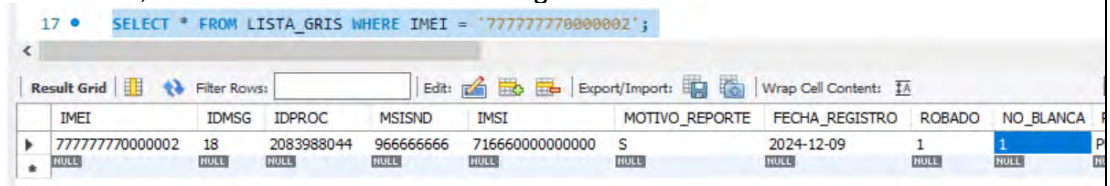
Validación, el IMEI no se encuentre en lista negra



Validación, el IMEI no se encuentra en lista blanca

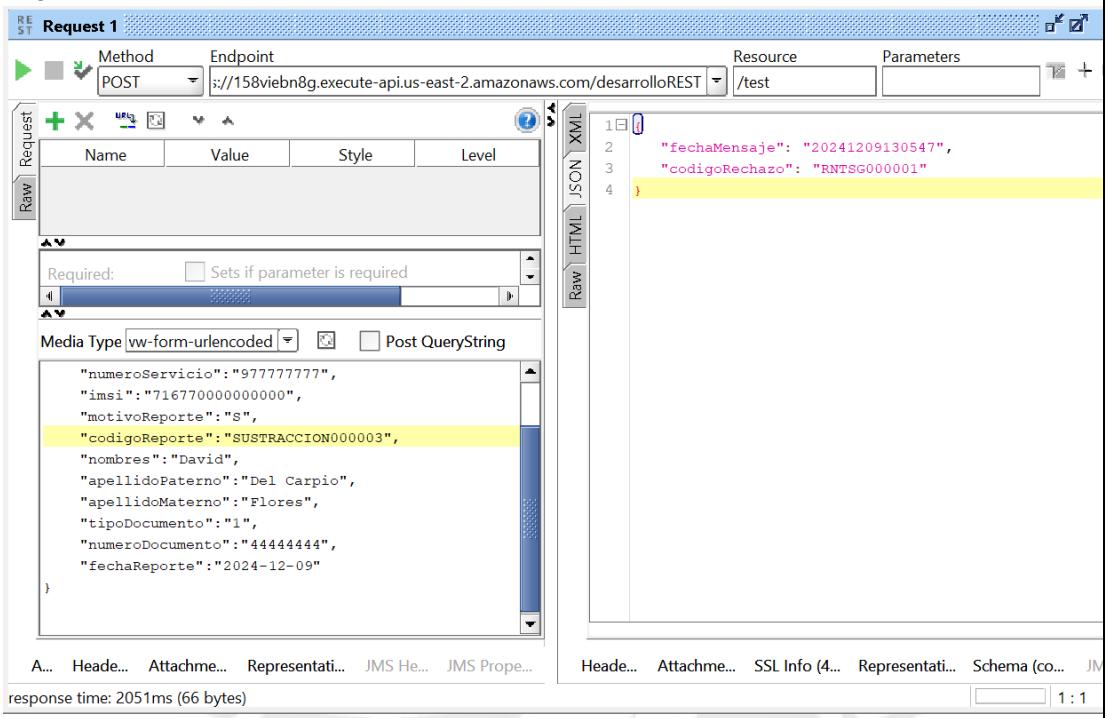


Validación, el IMEI se encuentra en lista gris

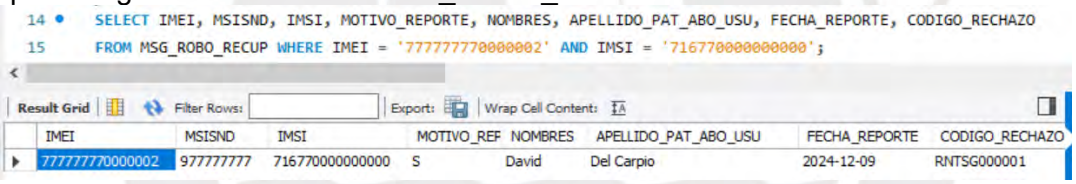


Entrada:	Reporte de Robo

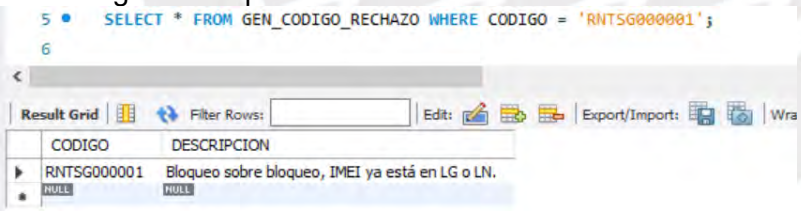
Luego de ello, se realiza la prueba en el API “API Bloqueo / Desbloqueo”, validando que el CEIR responde con los argumentos definidos en el caso de uso A-3.



Como se muestra en la respuesta se genera un código de error, este registro quedo registrado en la tabla MSG_ROBO_RECUP



En tabla GEN_CODIGO_RECHAZO se muestra la descripción del código de rechazo generado por el sistema

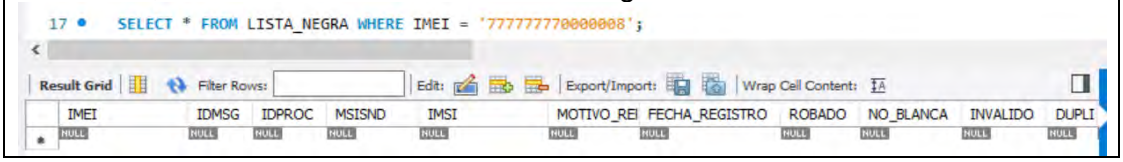


Acciones:	No aplica

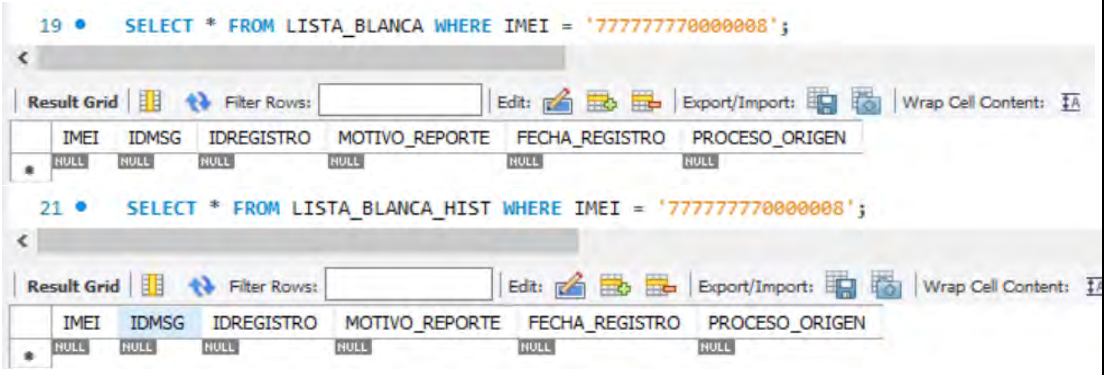
Caso de uso A-4

Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado **no** registrado en Lista Negra. IMEI reportado **no** registrado en Lista Blanca. IMEI reportado **no** registrado en Lista Gris

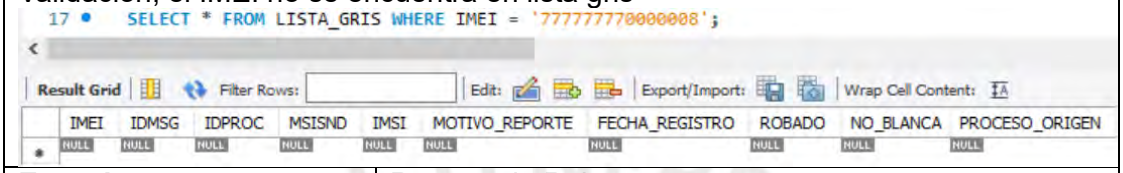
Validación, el IMEI no se encuentre en lista negra



Validación, el IMEI no se encuentra en lista blanca

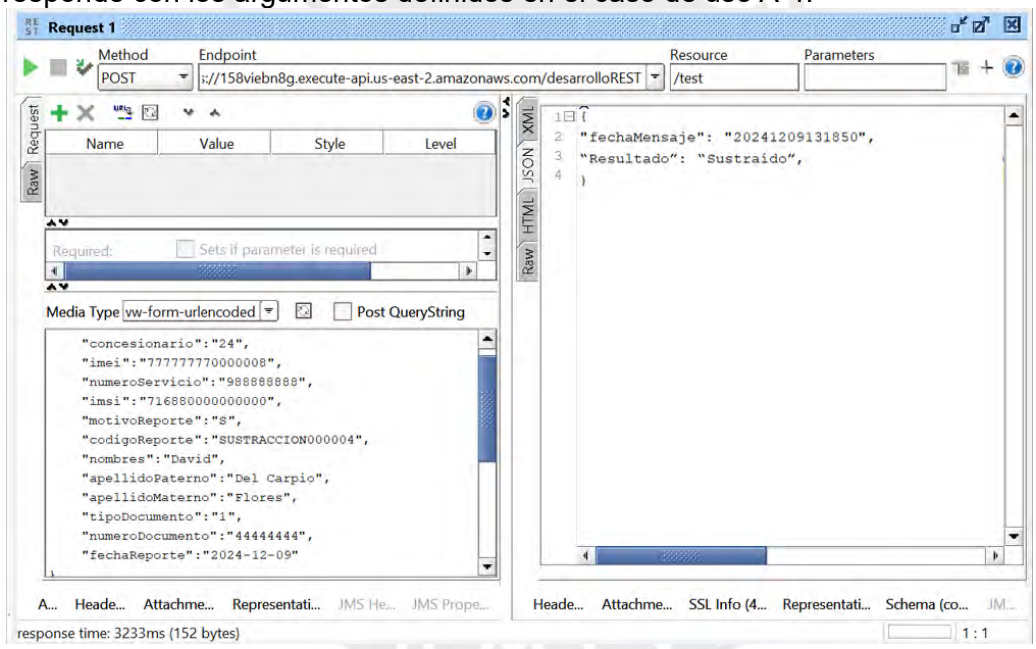


Validación, el IMEI no se encuentra en lista gris

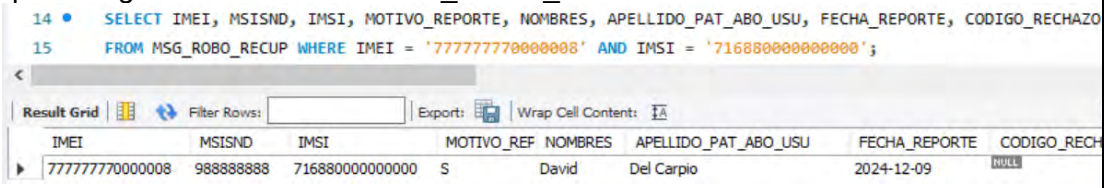


Entrada:	Reporte de Robo

Se realiza la prueba en el API “API Bloqueo / Desbloqueo”, validando que el CEIR responde con los argumentos definidos en el caso de uso A-4.

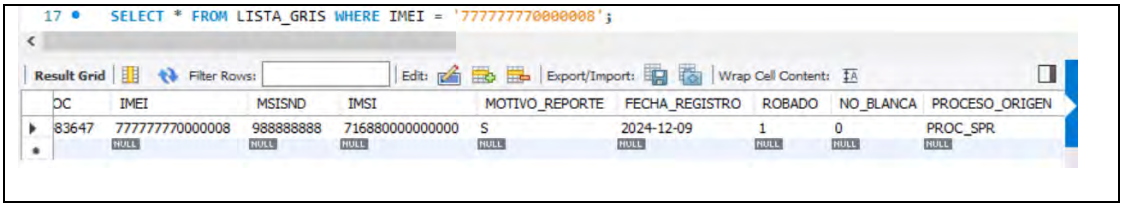


Como se muestra en la respuesta se genera una respuesta exitosa, este registro quedo registrado en la tabla MSG_ROBO_RECUP

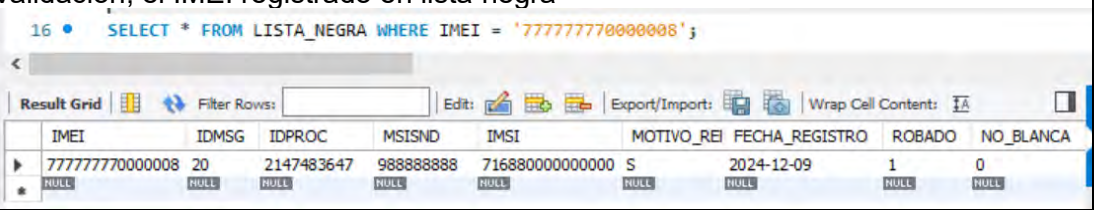
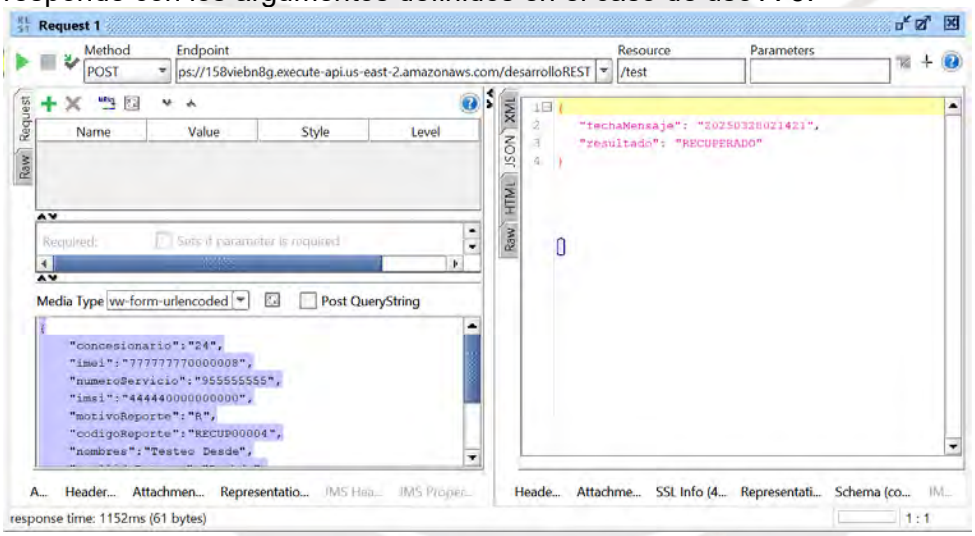
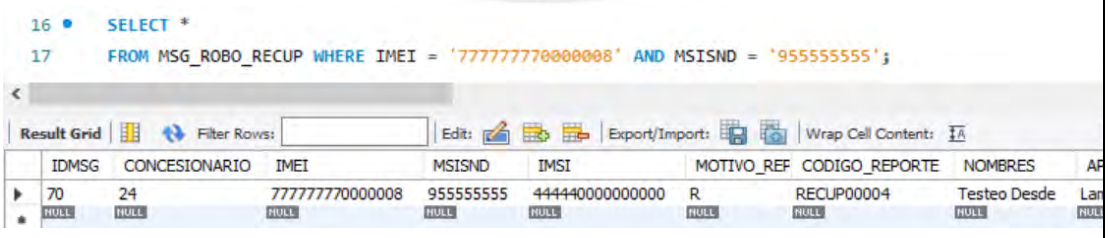


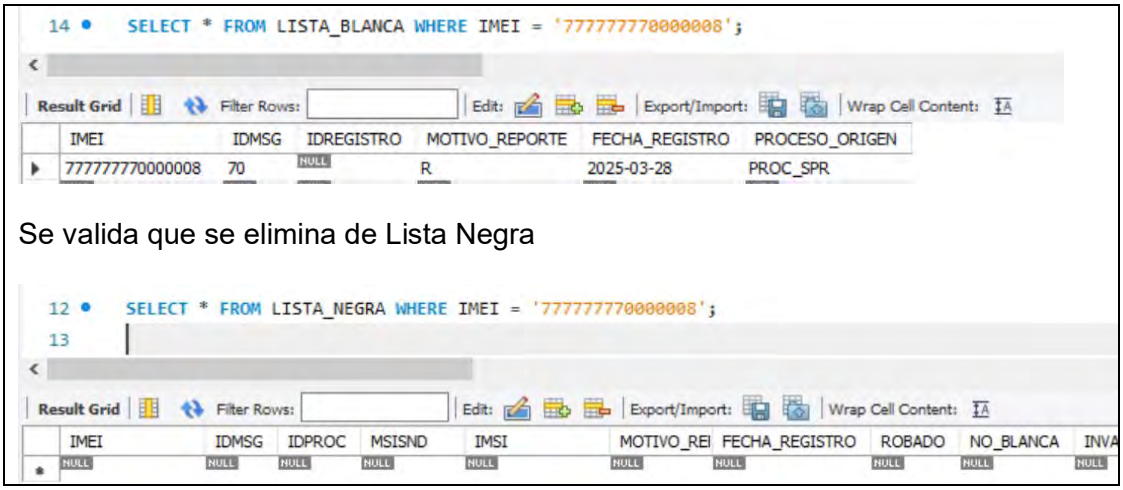
Acciones:	Registra el IMEI en Lista Gris

Se valida que el IMEI se registró en lista gris.



Caso de uso A-5

Recurso:	API Bloqueo / Desbloqueo
Precondición:	IMEI reportado registrado en Lista Negra.
Validación, el IMEI registrado en lista negra  	
Entrada:	Reporte de Recuperación
Se realiza la prueba en el API “API Bloqueo / Desbloqueo”, validando que el CEIR responde con los argumentos definidos en el caso de uso A-5.  Como se muestra en la respuesta se genera una respuesta exitosa, este registro quedo registrado en la tabla MSG_ROBO_RECUP  	
Acciones:	- Retorna IMEI a lista blanca siempre que se encuentre el IMEI en Lista_Blanca_Hist, si no se encuentra (Excepción 1: Genera Error) - Elimina IMEI de Lista Negra
Se valida que el IMEI retorna a Lista Blanca	

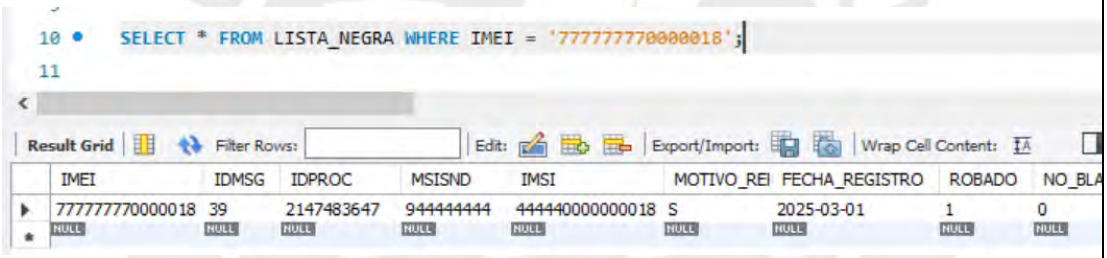


Se valida que se elimina de Lista Negra

Caso de uso B-1

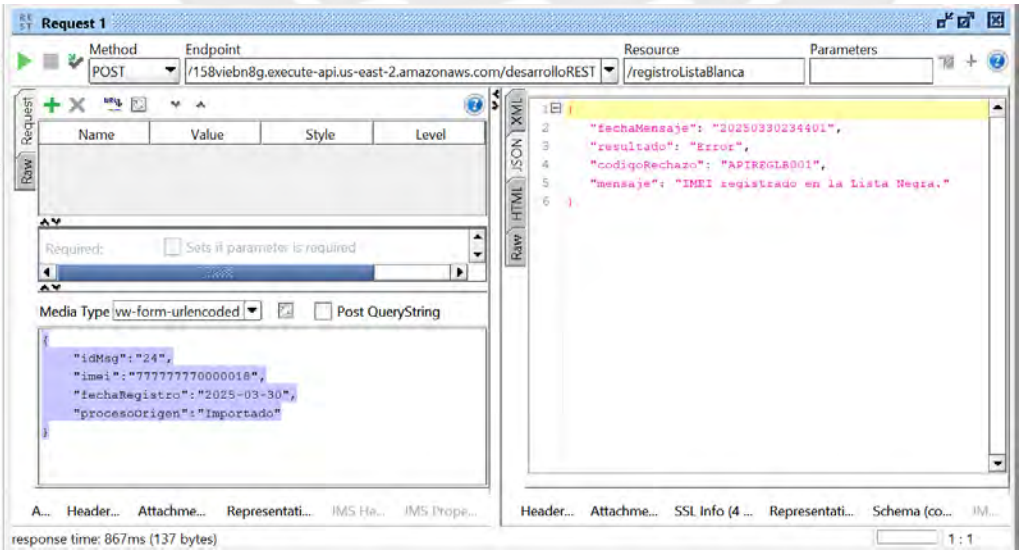
Recurso:	API Registro Lista Blanca
Precondición:	IMEI registrado en Lista Negra

Se valida como precondición que el IMEI exista Lista Negra del CEIR, se realizará la prueba con el IMEI “777777770000018”



Entrada:	Solicitud de Registro en Lista Blanca

Se valida que el CEIR responde con los argumentos definidos en el caso de uso B-1.

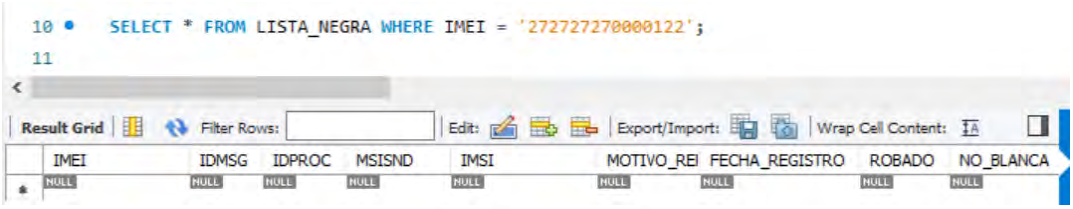


Acciones:	No aplica

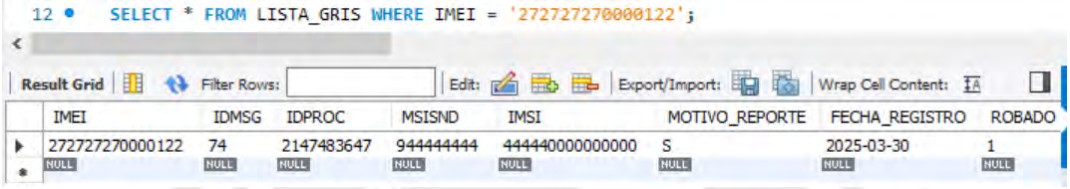
Caso de uso B-2

Recurso:	API Registro Lista Blanca
Precondición:	IMEI **no** registrado en Lista Negra IMEI registrado en Lista Gris como Robado

Se valida como precondición que el IMEI no se encuentre registrado en Lista Negra del CEIR.

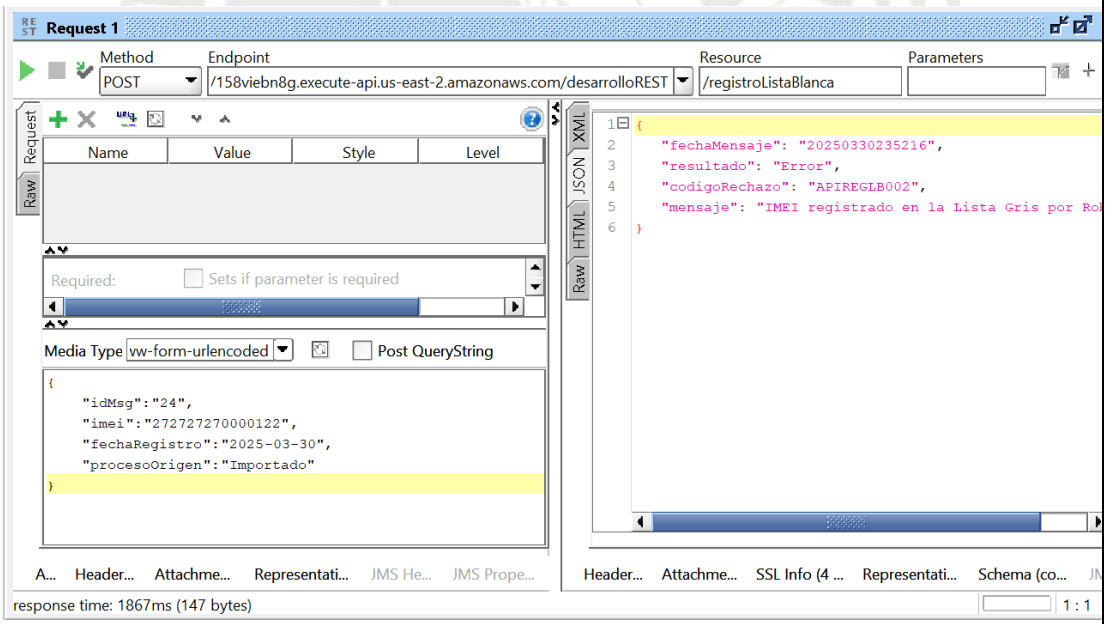


Se valida como precondición que el IMEI se encuentre registrado en Lista Gris del CEIR como Robado.



Entrada:	Solicitud de Registro en Lista Blanca

Se valida que el CEIR responde con los argumentos definidos en el caso de uso B-2.



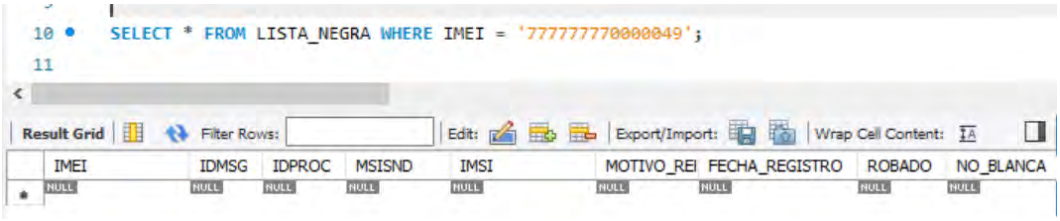
Acciones:	No aplica

Caso de uso B-3

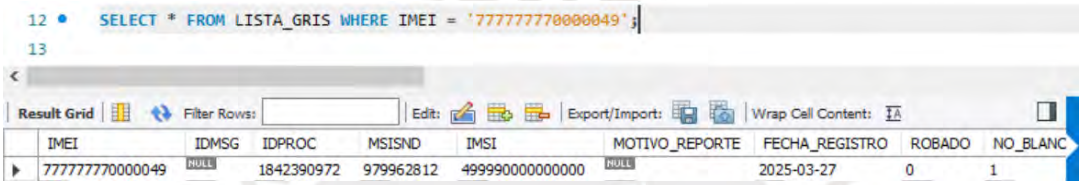
Recurso:	API Registro Lista Blanca
Precondición:	IMEI no registrado en Lista Negra

	IMEI registrado en Lista Gris como no registrado en Lista Blanca

Se valida como precondición que el IMEI no se encuentre registrado en Lista Negra del CEIR.

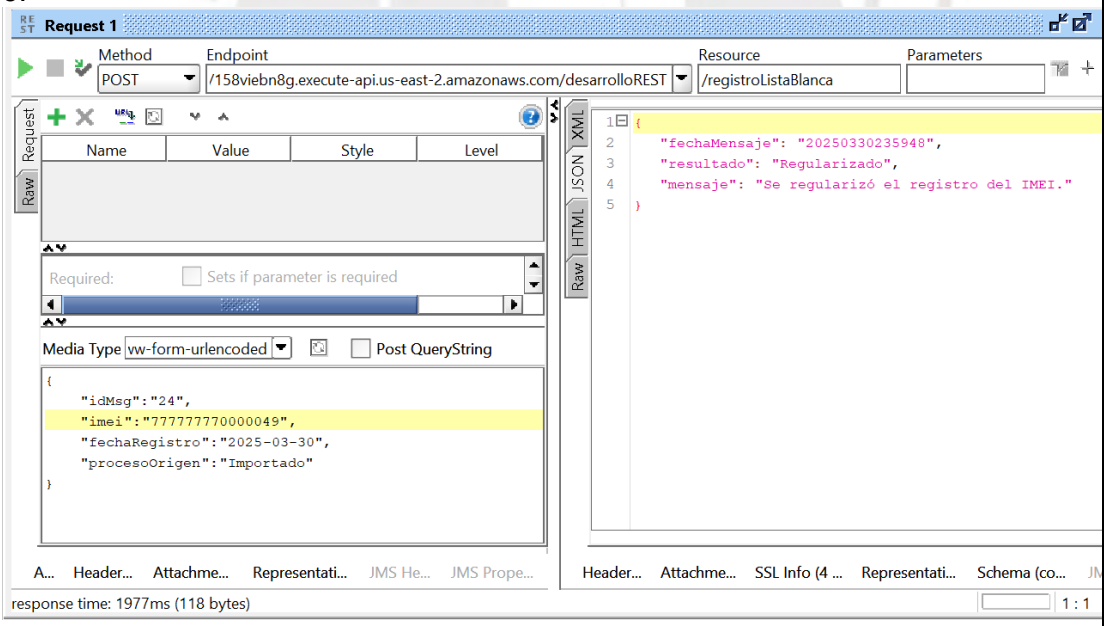


Se valida como precondición que el IMEI se encuentre registrado en Lista Gris del CEIR como No Registrado en Lista Blanca.



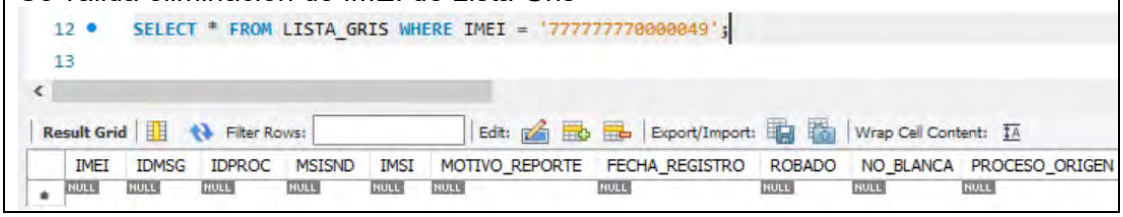
Entrada:	Solicitud de Registro en Lista Blanca

Se valida que el CEIR responde con los argumentos definidos en el caso de uso B-3.



Acciones:	- Se elimina el IMEI de Lista Gris - Se registra el IMEI en Lista Blanca

Se valida eliminación de IMEI de Lista Gris



Se valida el registro en Lista Blanca

```
12 •  SELECT * FROM LISTA_BLANCA WHERE IMEI = '777777770000049';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

IMEI	IDMSG	IDREGISTRO	MOTIVO_REPORTE	FECHA_REGISTRO	PROCESO_ORIGEN
777777770000049	24	NULL	NULL	2025-03-30	Importado
NULL	NULL	NULL	NULL	NULL	NULL

Caso de uso B-4

Recurso:	API Registro Lista Blanca
Precondición:	IMEI **no** registrado en Lista Negra. IMEI **no** registrado en Lista Gris. IMEI registrado en Lista Blanca

Se valida como precondición que el IMEI no se encuentre registrado en Lista Negra del CEIR.

```
10 •  SELECT * FROM LISTA_NEGRA WHERE IMEI = '777777770000099';
11
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDPROC	MSISND	IMSI	MOTIVO_REI	FECHA_REGISTRO	ROBADO	NO_BLANCA	INVALIDO
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Se valida como precondición que el IMEI NO se encuentre registrado en Lista Gris del CEIR.

```
10 •  SELECT * FROM LISTA_GRIS WHERE IMEI = '777777770000099';
11
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDPROC	MSISND	IMSI	MOTIVO_REPORTE	FECHA_REGISTRO	ROBADO	NO_BLANCA	PROCESO_ORIGEN
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Se valida como precondición que el IMEI se encuentre registrado en Lista Blanca del CEIR

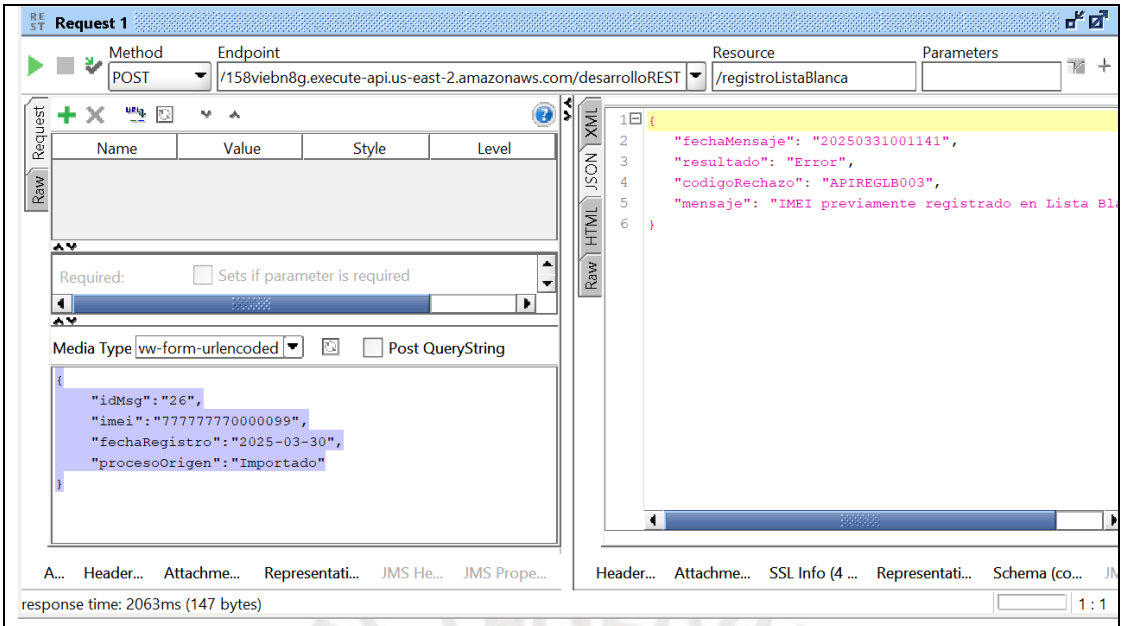
```
12 •  SELECT * FROM LISTA_BLANCA WHERE IMEI = '777777770000099';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Co

IMEI	IDMSG	IDREGISTRO	MOTIVO_REPORTE	FECHA_REGISTRO	PROCESO_ORIGEN
777777770000099	26	NULL	NULL	2025-03-31	Importado
NULL	NULL	NULL	NULL	NULL	NULL

Entrada:	Solicitud de Registro en Lista Blanca

Se valida que el CEIR responde con los argumentos definidos en el caso de uso B-4.

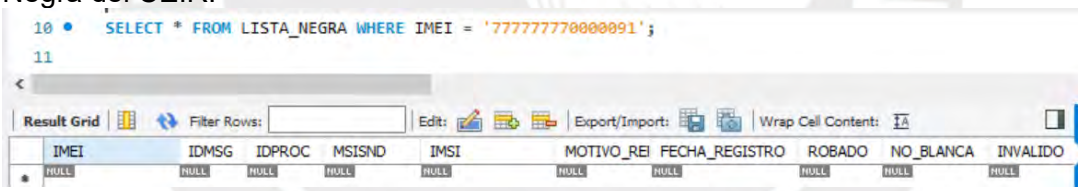


Acciones:	No aplica

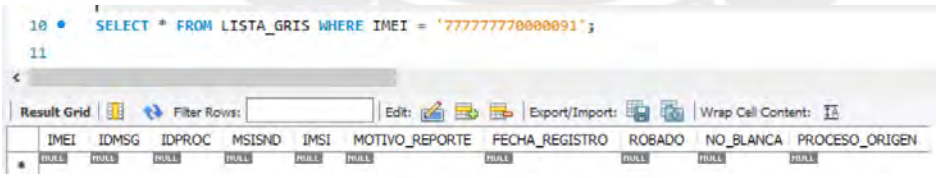
Caso de uso B-5

Recurso:	API Registro Lista Blanca
Precondición:	IMEI **no** registrado en Lista Negra IMEI **no** registrado en Lista Gris IMEI **no** registrado en Lista Blanca

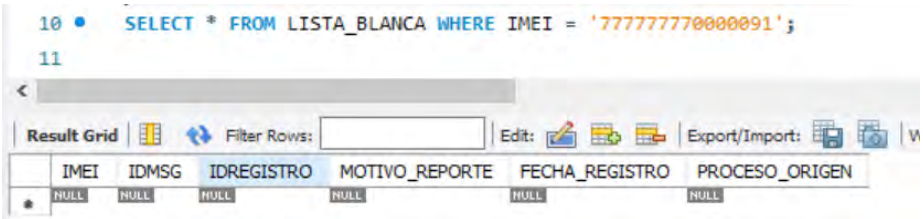
Se valida como precondición que el IMEI no se encuentre registrado en Lista Negra del CEIR.



Se valida como precondición que el IMEI no se encuentre registrado en Lista Gris del CEIR.

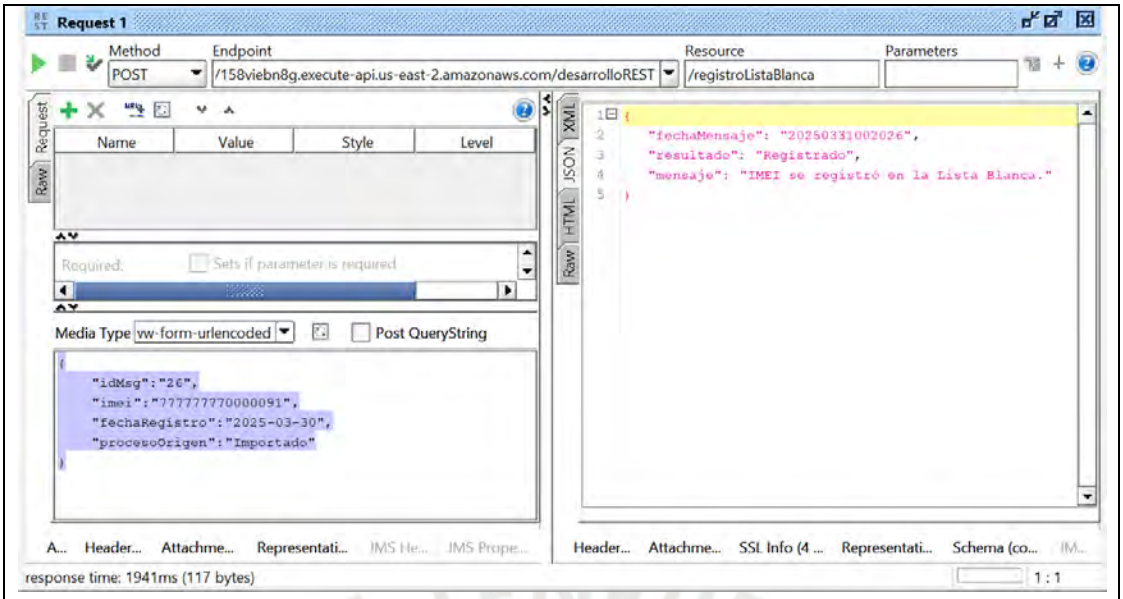


Se valida como precondición que el IMEI no se encuentre registrado en Lista Blanca del CEIR.



Entrada:	Solicitud de Registro en Lista Blanca

Se valida que el CEIR responde con los argumentos definidos en el caso de uso B-5.



Acciones:	Se registra el IMEI en Lista Blanca

Se valida el registro en Lista Blanca

```
SELECT * FROM LISTA_BLANCA WHERE IMEI = '777777770000091';
```

Result Grid

IMEI	IDMSG	IDREGISTRO	MOTIVO_REPORTE	FECHA_REGISTRO	PROCESO_ORIGEN
777777770000091	26	NULL	NULL	2025-03-31	Importado
NULL	NULL	NULL	NULL	NULL	NULL

Caso de uso D-1

Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI registrado en Lista Negra del CEIR

Se valida como precondición que el IMEI exista Lista Negra del CEIR, se realizará la prueba con el IMEI “987654321987654”

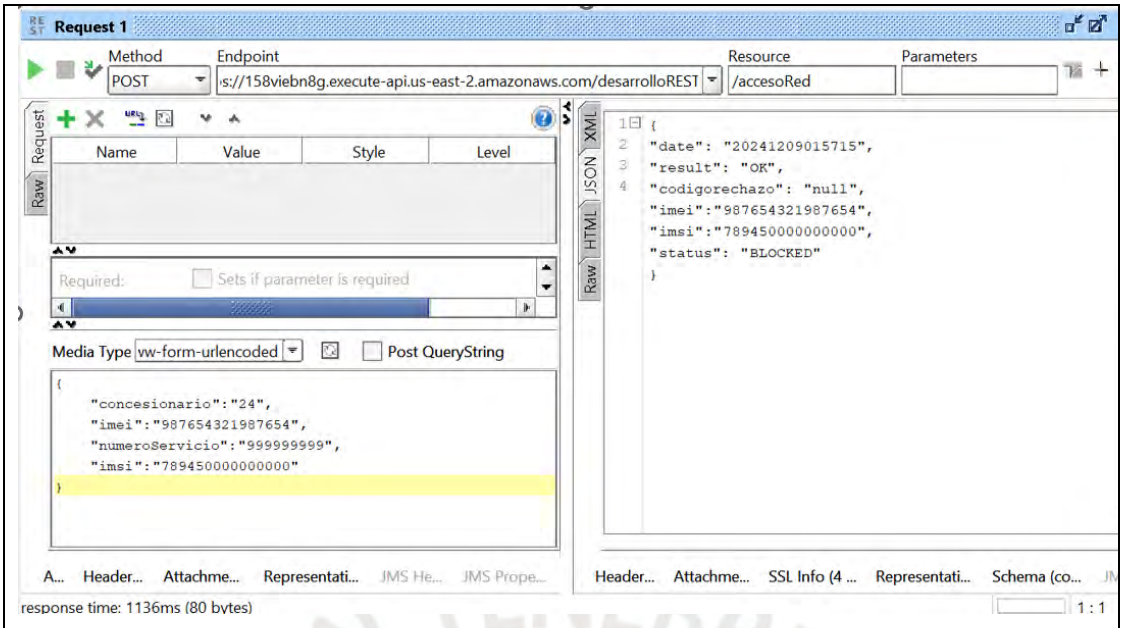
```
SELECT * FROM LISTA_NEGRA WHERE IMEI IN (
'987654320000000',
'987654321987654');
```

Result Grid

IMEI	MSISND	IMSI	MOTIVO_REPORTE	FECHA_REGISTRO	ROBADO	NO_BLANCA	INVALIDO	DUPLI
987654320000000	999999999	789450000000000	P	2024-12-06	1	0	0	0
987654321987654	999999999	789456123789456	S	2024-12-06	1	1	0	0

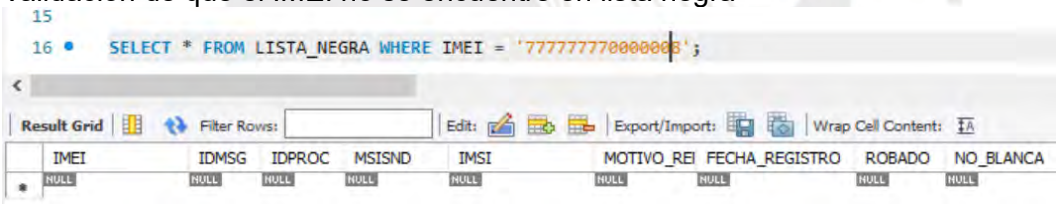
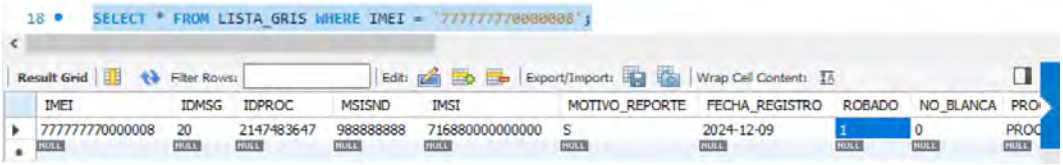
Entrada:	Respuesta de Bloqueo del CEIR

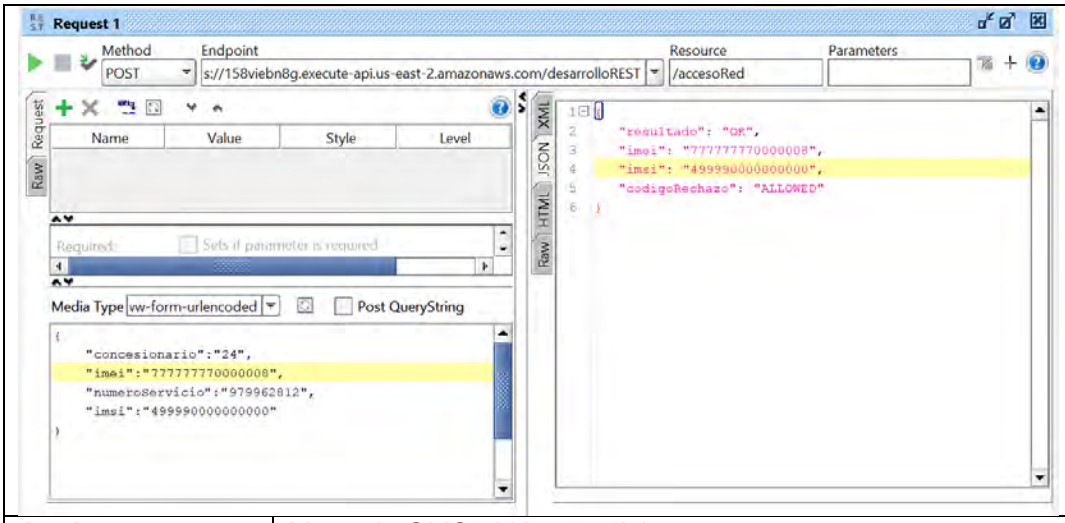
Se valida que el CEIR responde con los argumentos definidos en el caso de uso D-1.



Acciones:	No aplica

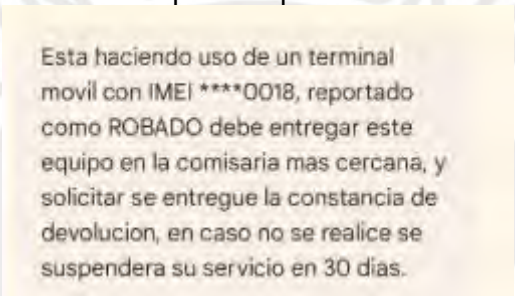
Caso de uso D-2

Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI registrado en Lista Gris como Robado
Validación de que el IMEI no se encuentre en lista negra  Validación que el IMEI se encuentra en Lista Gris 	
Entrada:	Respuesta de CEIR IMEI en Trazabilidad
Se valida que el CEIR responde con los argumentos definidos en el caso de uso D-2.	

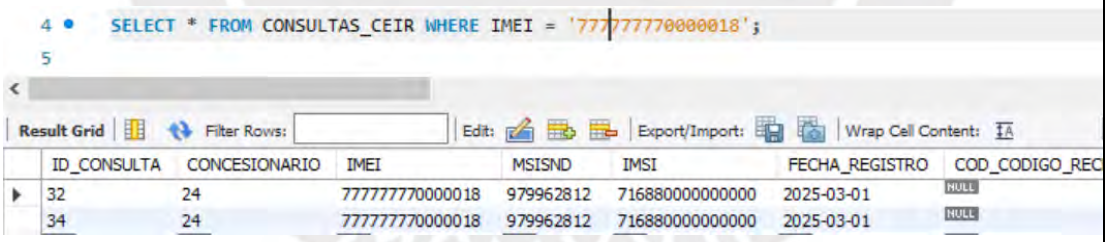


Acciones:	Mensaje SMS al Nro servicio Registrar evento en tabla de trazabilidad.

Se valida envío de SMS de IMEI reportado previamente como robado



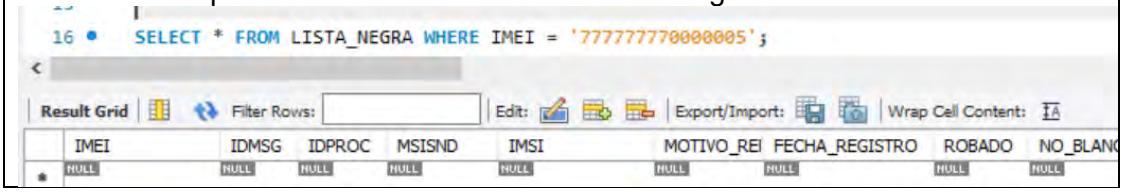
Se valida registro de trazabilidad



Caso de uso D-3

Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI registrado en Lista Gris como No Registrado Lista Blanca

Validación de que el IMEI no se encuentre en lista negra



Validación que el IMEI se encuentra en Lista Gris

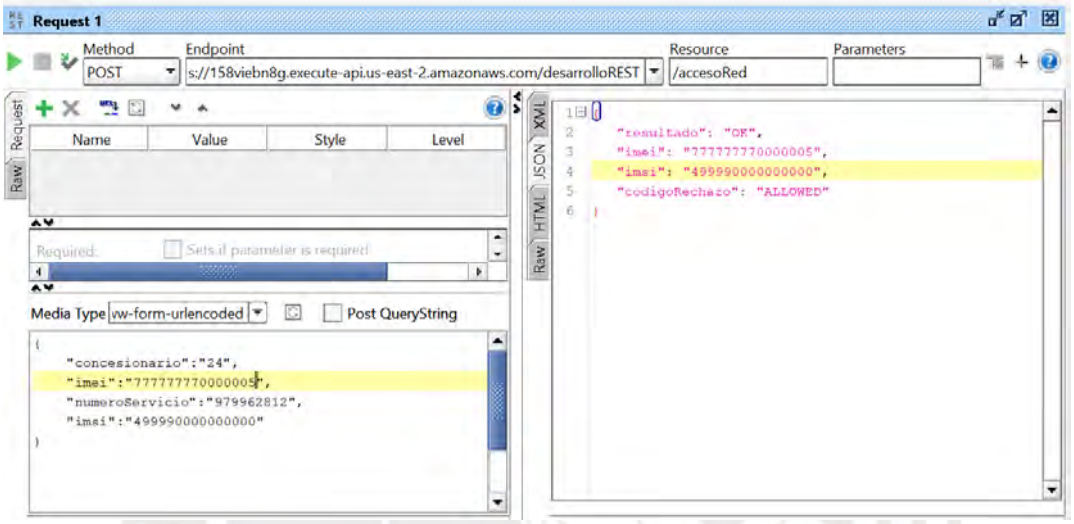
```
17
18 •  SELECT * FROM LISTA_GRIS WHERE IMEI = '777777770000005';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDPROC	MSISND	IMSI	MOTIVO_REPORTE	FECHA_REGISTRO	ROBADO	NO_BLANCA
777777770000005	7	2147483647	944444444	444440000000000		2024-12-06	0	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

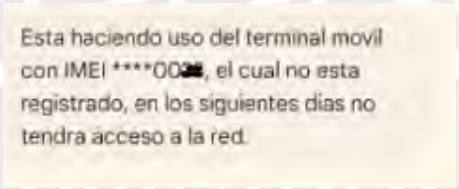
Entrada:	Respuesta de CEIR IMEI en Trazabilidad

Se valida que el CEIR responde con los argumentos definidos en el caso de uso D-3.



Acciones:	Mensaje SMS al Nro servicio Registrar evento en tabla de trazabilidad.

Se valida envío de SMS de IMEI no registrado en Lista Blanca



Se valida registro de trazabilidad

```
4 •  SELECT * FROM CONSULTAS_CEIR WHERE IMEI = '777777770000005';
5
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Co

ID_CONSULTA	CONCESIONARIO	IMEI	MSISND	IMSI	FECHA_REGISTRO
8	24	777777770000005	979962812	499990000000000	2024-12-13

Caso de uso D-4

Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI **no** registrado en Lista Gris del CEIR

	IMEI registrado en Lista Blanca

Validación de que el IMEI no se encuentre en lista negra

```
7 •  SELECT * FROM LISTA_NEGRA WHERE IMEI = '777777770000001';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDPROC	MSISND	IMSI	MOTIVO_REPORTE	FECHA_REGISTRO	ROBADO	NO_BLANCA	INVALIDO	DUPLICAD
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Validación que el IMEI no se encuentra en Lista Gris

```
9 •  SELECT * FROM LISTA_GRIS WHERE IMEI = '777777770000001';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDPROC	MSISND	IMSI	MOTIVO_REPORTE	FECHA_REGISTRO	ROBADO	NO_BLANCA	PROCESO_ORIGEN
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Validación que el IMEI se encuentra en Lista Blanca

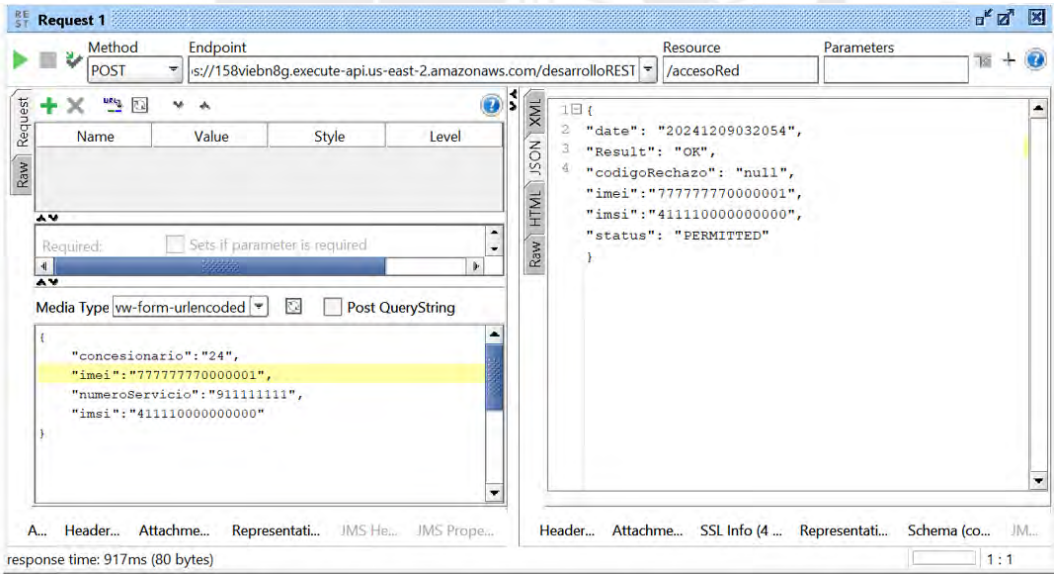
```
11 •  SELECT * FROM LISTA_BLANCA WHERE IMEI = '777777770000001';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IMEI	IDMSG	IDREGISTRO	MOTIVO_REPORTE	FECHA_REGISTRO	PROCESO_ORIGEN
777777770000001	100	2	NULL	2024-12-06	PROC_TERM_IMP

Entrada:	Respuesta de CEIR IMEI en Permitido

Se valida que el CEIR responde con los argumentos definidos en el caso de uso D-4.



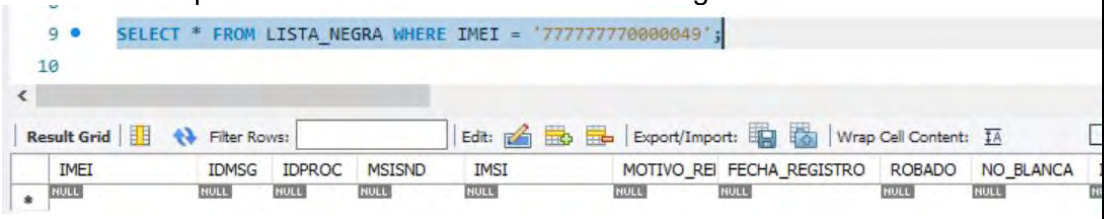
Acciones:	No Aplica

Caso de uso D-5

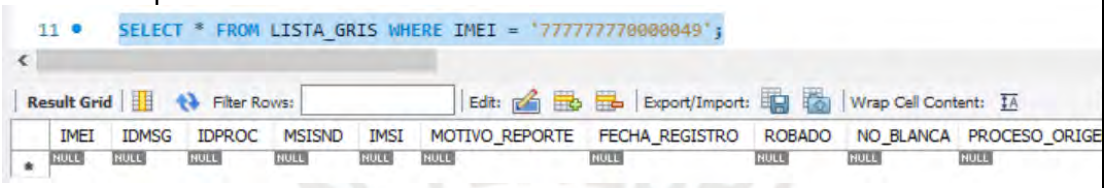
Recurso:	API Gestión de acceso a la red
Precondición:	IMEI no registrado en EIR IMEI **no** registrado en Lista Negra del CEIR IMEI **no** registrado en Lista Gris del CEIR

	IMEI **no** registrado en Lista Blanca

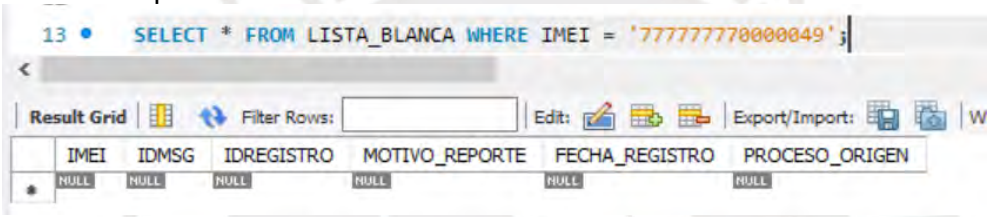
Validación de que el IMEI no se encuentre en lista negra



Validación que el IMEI no se encuentra en Lista Gris

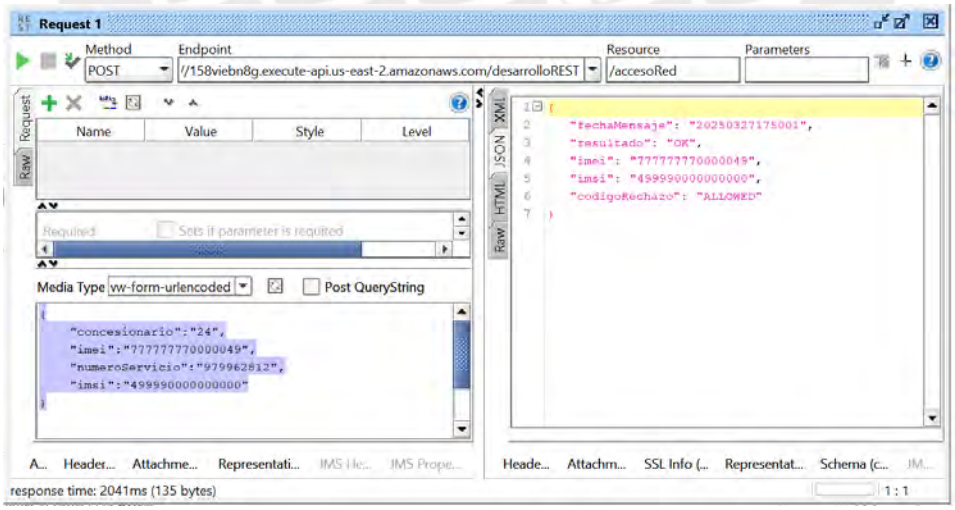


Validación que el IMEI no se encuentra en Lista Blanca



Entrada:	Respuesta de CEIR IMEI no registrado

Se valida que el CEIR responde con los argumentos definidos en el caso de uso D-5



Acciones:	- Mensaje SMS al Nro servicio “Está haciendo uso del terminal móvil no registrado, en los siguientes días no tendrá acceso a la red” -Registrar evento en tabla de trazabilidad.

Se valida envío de SMS de IMEI no registrado en Lista Blanca

Esta haciendo uso del terminal movil con IMEI ****00[redacted], el cual no esta registrado, en los siguientes dias no tendra acceso a la red.

Se valida registro de trazabilidad

```
4 •  SELECT * FROM CONSULTAS_CEIR WHERE IMEI = '777777770000049';
5
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ID_CONSULTA	CONCESIONARIO	IMEI	MSISND	IMSI	FECHA_REGISTRO	COD_CODIGO_REC
41	24	777777770000049	979962812	499990000000000	2025-03-27	NULL
42	24	777777770000049	979962812	499990000000000	2025-03-27	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

ANEXO 2: Código en Python de las APIS desplegadas en servicios Lambda

I) API Bloqueo / Desbloqueo

```
import mysql.connector
import misFunciones as mf
from datetime import datetime
from mysql.connector import Error
import json
import random
import os

tac_invalidos = ['00000000', '11111111', '12345678']

def lambda_handler(event, context):

    global connection
    '''
    Proceso de reporte SPR
    '''
    imei = event.get('imei')
    tac = imei[:8]

concesionario,numeroServicio,imsi,motivoReporte,codigoReporte,nombres,ape
llidoPaterno,apellidoMaterno,\
    tipoDocumento,numeroDocumento,fechaReporte = \
        event.get('concesionario'),event.get('numeroServicio'),event.get(
'imsi'),event.get('motivoReporte'),\
        event.get('codigoReporte'),event.get('nombres'),event.get('apelli
doPaterno'),event.get('apellidoMaterno'),\
        event.get('tipoDocumento'),event.get('numeroDocumento'),event.get
('fechaReporte')

    if not imei:
        codigoRechazo = 'RNTSG000004'
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte,
            codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
            tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo)
        mensaje = 'Falta o está vacío el campo IMEI en la solicitud.'
        return {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'resultado': 'ERROR',
        'codigoRechazo':codigoRechazo,
```

```
        'descripción': mensaje
        }
    elif len(imei) != 15 or not imei.isdigit():
        codigoRechazo = 'RNTSG000005'
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo)
        mensaje = 'IMEI - longitud incorrecta o valor inválido.'
        return {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'resultado': 'ERROR',
        'codigoRechazo':codigoRechazo,
        'descripción': mensaje
        }
    elif tac in tac_invalidos:
        codigoRechazo = 'RNTSG000006'
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo)
        mensaje = 'TAC es inválido.'
        return {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'resultado': 'ERROR',
        'codigoRechazo':codigoRechazo,
        'descripción': mensaje
        }

    try:
        if motivoReporte == 'R':
            #Proceso de recuperación de un IMEI
      if mf.inListaNegra(imei) != False or mf.inListaGris(imei) != False:

                if mf.inListaNegra(imei) != False:
                    print("La causa fue inListaNegra.")
                    if mf.motivoReporteLN(imei) != True:
                        print('IMEI solo está por Robo')


output = mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio,
imsi,    motivoReporte,    codigoReporte,    nombres,    apellidoPaterno,
apellidoMaterno, tipoDocumento, numeroDocumento, fechaReporte)

                        return {
                        'fechaMensaje':
datetime.now().strftime('%Y%m%d%H%M%S'),
                        'resultado': 'RECUPERADO'
                        }
```

```
                    else:
                        codigoRechazo = 'RNTSG000003'
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo)
mensaje = 'El IMEI tiene más razones de bloqueo, no es posible
desbloquear.'
                        return {
                        'fechaMensaje':
datetime.now().strftime('%Y%m%d%H%M%S'),
                        'resultado': 'ERROR',
                        'codigoRechazo':codigoRechazo,
                        'descripción': mensaje
                        }

                if mf.inListaGris(imei) != False:
                    print("La causa fue en Lista Gris.")

                    if mf.motivoReporteLG(imei) != True:
                        print('IMEI solo está por Robo')

mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte)
                        return {
                        'fechaMensaje':
datetime.now().strftime('%Y%m%d%H%M%S'),
                        'resultado': 'RECUPERADO'
                        }

                    else:
                        codigoRechazo = 'RNTSG000003'
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo)
mensaje = 'El IMEI tiene más razones de bloqueo, no es posible
desbloquear.'
                        return {
                        'fechaMensaje':
datetime.now().strftime('%Y%m%d%H%M%S'),
                        'resultado': 'ERROR',
                        'codigoRechazo':codigoRechazo,
                        'descripción': mensaje
                        }


            else:
                codigoRechazo = 'RNTSG000002'
```

```
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo)

mensaje = 'Se está intentando desbloquear un ETM que no está en la Lista
Negra o Lista Gris.'
                return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'resultado': 'ERROR',
                'codigoRechazo':codigoRechazo,
                'descripción': mensaje
                }


        else:
            #Comprobar si está en LN o LG
            if mf.inListaNegra(imei) == True or mf.inListaGris(imei)==
True:
    #Error de bloqueo sobre bloqueo y solo se debe suspender el servicio
                codigoRechazo='RNTSG000001'
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte,codigoRechazo)
mensaje = 'Bloqueo sobre bloqueo, IMEI ya está en LG o LN.'
                return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'resultado': 'ERROR',
                'codigoRechazo':codigoRechazo,
                'descripción': mensaje
                }

            else:
                #Registrar en la lista GRIS
mf.insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte)
                return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'resultado': 'SUSTRAIDO'
                }

    except Error as e:
        # En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e}"
        }

    finally:
```

```
        # Cerrar la conexión si está abierta
        mf.cerrar_conexion()
        print("Conexión a MySQL cerrada")
```

Función de registro de información en la base de datos: MisFunciones.py para el API Bloqueo Desbloqueo

```
import mysql.connector
from datetime import datetime
from mysql.connector import Error
import json
import random
import os

connection = None
def conectar_db():
    global connection
    try:
        connection = mysql.connector.connect(
            host= os.getenv('DATABASE_HOST'),
            database= os.getenv('DATABASE_PRO'),
            user= os.getenv('DATABASE_USER'),
            password= os.getenv('DATABASE_PASS')
        )
        if connection.is_connected():
            print("Conexión exitosa a la base de datos.")
    except mysql.connector.Error as err:
        print(f"Error al conectar a la base de datos: {err}")

def cerrar_conexion():
    global connection
    if connection and connection.is_connected():
        connection.close()
        print("Conexión cerrada.")

def inListaBlancaHist(imei):
    '''
    Valida si el IMEI existe en la Lista Blanca Histórica.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
```

```
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_BLANCA_HIST WHERE imei =
%s", (imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False


def inListaNegra(imei):
    '''
    Valida si el IMEI existe en la Lista Negra.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_NEGRA WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def inListaGris(imei):
    '''
    Valida si el IMEI existe en la Lista Gris.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la Lista Gris,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
```

```
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_GRIS WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def inListaBlanca(imei):
    '''
    Valida si el IMEI existe en la Lista Blanca.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la Lista Blanca,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_BLANCA WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def nTerminalesImp(imei):
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor()
        query = """
            SELECT
                CASE
WHEN COUNT(DISTINCT IMEI1) + COUNT(DISTINCT IMEI2) = 2 THEN 2
WHEN COUNT(DISTINCT IMEI1) + COUNT(DISTINCT IMEI2) = 1 THEN 1
                    ELSE 0
                END AS resultado
            FROM
                TERM_IMPORT
```

```
            WHERE
                IMEI1 = %s OR IMEI2 = %s;
            """
        cursor.execute(query, (imei, imei))
        resultado = cursor.fetchone()
        if resultado:
            return resultado[0]  # Devolver el resultado (puede ser 2 ó
1)
        else:
            return 0
    except Exception as e:
        print(f"Error al verificar IMEI: {e}")
        return None

def insertarRoboRecuperacion(concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, nombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo=None):
    try:
        if connection is None or not connection.is_connected():
            conectar_db()

        cursor = connection.cursor()
        query = """
INSERT INTO MSG_ROBO_RECUP (IDMSG, CONCESIONARIO, IMEI, MSISND, IMSI,
MOTIVO_REPORTE, CODIGO_REPORTE, NOMBRES, APELLIDO_PAT_ABO_USU,
APELLIDO_MAT_ABO_USU, TIPO_DOC_ABONADO, NRO_DOC_ABONADO, FECHA_REPORTE,
FECHA_REGISTRO, CODIGO_RECHAZO)
VALUES (NULL, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
CURRENT_TIMESTAMP, %s)
        """
        params = (concesionario, imei, numeroServicio, imsi,
motivoReporte, codigoReporte, ombres, apellidoPaterno, apellidoMaterno,
tipoDocumento, numeroDocumento, fechaReporte, codigoRechazo if
codigoRechazo is not None else None)

        cursor.execute(query, params)
        connection.commit()
        print("Registro histórico en Recuperación Robo guardado
exitosamente.")


#Query2 en caso el IMEI no estaba en Lista Gris y no tuvo códigoRechazo.
        idmsg = cursor.lastrowid
        esta = inListaBlanca(imei)
        print(esta,' : ',idmsg)# AQUI ESTA EL IDMSG OBTENIDO

        if motivoReporte == 'R' and codigoRechazo is None:
            if connection is None or not connection.is_connected():
                conectar_db()
```

```
            if idmsg is None:
                raise Exception("No se pudo obtener el IDMSG generado")

#Verificar que esté en la Lista Gris o Negra y si está o no en la
BlancaHist.
            cantidadDeIMei = nTerminalesImp(imei)
            cursor = connection.cursor()
            #GRis y Blanca_hist
            if inListaGris(imei) and inListaBlancaHist(imei):
                print("**ESTA EN GRIS Y BLANCA_HIST**")
                '''
Sacar de GRIS
ELIMINAR DE LA LISTA BLANCA HISTORICA (Dejar Registro en ListaBlanca HIST
)
                Devolver el IMEI a Lista Blanca
                '''
                if cantidadDeIMei == 1:
                    deleteListaGris(imei)
                    deleteListaBlancaHist(imei)
                    insertListaBlanca(imei,idmsg,motivoReporte)
mensaje = 'Se ha recuperado el IMEI, se ha regresado a la Lista Blanca. 1
Terminal registrada en el IMPORT.'
print('Recuperación de IMEI en GRIS y en Lista Blanca con 1 IMEI en
IMPORT.')
                elif cantidadDeIMei == 2:
                    deleteListaGris(imei)
                    deleteListaBlancaHist(imei)
insertListaBlanca(imei,idmsg,motivoReporte)
mensaje = 'Se ha recuperado el IMEI, se ha regresado a la Lista Blanca. 2
terminales registradas en el IMPORT.'
print('Recuperación de IMEI en GRIS y en Lista Blanca con 2 IMEI en
IMPORT.')

                else:
                    print('No se encontró IMEI en EMP_IMPORT')
                connection.commit()

                print('Existe en la blanca HIST.')
            elif inListaGris(imei) and inListaBlancaHist(imei)==False:
                '''
                Sacar de GRIS
                '''
                if cantidadDeIMei == 1:
                    deleteListaGris(imei)
mensaje = 'Se ha recuperado el IMEI, NO estaba en la Lista Blanca
Anteriormente. 1 Terminal registrada en el IMPORT.'
print('Recuperación de IMEI en GRIS y NO Lista Blanca con 1 IMEI en
IMPORT.')
                elif cantidadDeIMei == 2:
```

```
                    deleteListaGris(imei)
mensaje = 'Se ha recuperado el IMEI, NO estaba en la Lista Blanca
Anteriormente. 2 terminales registradas en el IMPORT.'
print('Recuperación de IMEI en GRIS y NO Lista Blanca con 2 IMEI en
IMPORT.')

                else:
                    print('No se encontró IMEI en EMP_IMPORT')
                connection.commit()

            elif inListaNegra(imei) and inListaBlancaHist(imei):
                print("**ESTA EN NEGRA Y BLANCA_HIST**")
                '''
Sacar de NEGRA ELIMINAR DE LISTA BLANCA HISTORICA (Dejar Registro en
ListaBlanca HIST)
Devolver el IMEI a Lista Blanca
                '''
                if cantidadDeIMei == 1:

                    deleteListaNegra(imei)
                    deleteListaBlancaHist(imei)
                    insertListaBlanca(imei,idmsg,motivoReporte)
mensaje = 'Se ha recuperado el IMEI, se ha regresado a la Lista Blanca. 1
Terminal registrada en el IMPORT.'
print('Recuperación de IMEI en NEGRA y en Lista Blanca con 1 IMEI en
IMPORT.')


                elif cantidadDeIMei == 2:
                    deleteListaNegra(imei)
                    deleteListaBlancaHist(imei)
                    insertListaBlanca(imei,idmsg,motivoReporte)
mensaje = 'Se ha recuperado el IMEI, se ha regresado a la Lista Blanca. 2
Terminales registradas en el IMPORT.'
print('Recuperación de IMEI en NEGRA y en Lista Blanca con 2 IMEI en
IMPORT.')

                else:
mensaje = 'El IMEI no puede volver a la lista blanca porque no está en la
tabla de importados.'
print('No se encontró IMEI en EMP_IMPORT')
connection.commit()

            elif inListaNegra(imei) and inListaBlancaHist(imei) == False:
                '''
                Sacar de NEGRA
                '''
                if cantidadDeIMei == 1:
                    deleteListaNegra(imei)
```

```
mensaje = 'Se ha recuperado el IMEI, NO estaba en la Lista Blanca
Anteriormente. 1 Terminal registrada en el IMPORT.'
print('Recuperación de IMEI en NEGRA y NO Lista Blanca con 1 IMEI en
IMPORT.')
                elif cantidadDeIMei == 2:
                    deleteListaNegra(imei)
mensaje = 'Se ha recuperado el IMEI, NO estaba en la Lista Blanca
Anteriormente. 2 Terminales registradas en el IMPORT.'
print('Recuperación de IMEI en NEGRA y NO Lista Blanca con 2 IMEI en
IMPORT.')

                else:
                    print('No se encontró IMEI en EMP_IMPORT')
                connection.commit()
            return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'mensaje': mensaje
                }

#Para registros que no estén en la Lista Blanca y sean S o P
        if codigoRechazo is None and inListaBlanca(imei) == False and
motivoReporte in('S','P'):
            print("entré al if de NONE")
            if idmsg is None:
                raise Exception("No se pudo obtener el IDMSG generado")

# Generar un IDPROC
            idproc = ''.join([str(random.randint(0, 9)) for _ in
range(10)])

            query2 = """
INSERT INTO LISTA_GRIS (IMEI, IDMSG, IDPROC, MSISND, IMSI,
MOTIVO_REPORTE, FECHA_REGISTRO, ROBADO, NO_BLANCA, PROCESO_ORIGEN)
VALUES (%s, %s, %s, %s, %s, %s, CURRENT_TIMESTAMP, %s, %s, %s)
            """
#VALIDAR SI ESTA EN BLANCA CON VALORES 1 o =, también validar si es S o P
            params2 = (
                imei, idmsg, idproc, numeroServicio, imsi, motivoReporte,
                '1', '0', 'PROC_SPR'  # ROBADO = '1' y PROCESO_ORIGEN =
'PROC_SPR'
            )

# Ejecutar la inserción de la segunda tabla
            cursor.execute(query2, params2)

# Si ambas inserciones fueron exitosas, realizar commit de la transacción
            connection.commit()
            print('INSERTÉ EN GRIS')
```

```
#Para los que estén en Lista Blanca, sin distinguir S P o R
        elif codigoRechazo is None and inListaBlanca(imei) == True and
motivoReporte in('S','P'):
            cantidadLB = nTerminalesImp(imei)
            print('Cantidad en LB:', cantidadLB)
            if cantidadLB == 2:
                print('2IMEIS en LB')

                deleteListaBlanca(imei)

#FUNCIÓN PARA HACER EL DELETE EN LA LISTA BLANCA.
                insertListaBlancaHist(imei,idmsg,motivoReporte)
            insertListaGris(imei,idmsg,numeroServicio,imsi,motivoReporte)


                mensaje = {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'codigoRechazo':'Tiene 2 IMEIS'
                }
                return mensaje
            elif cantidadLB == 1:
                print('1 IMEIS en LB')
                deleteListaBlanca(imei)
                insertListaBlancaHist(imei,idmsg,motivoReporte)
            insertListaGris(imei,idmsg,numeroServicio,imsi,motivoReporte)
                return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'codigoRechazo':'Tiene 1 IMEI'
                }
            else:
                print('NO EXISTE')
                return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'cantidad':cantidadLB,
                'codigoRechazo':'No existe en la LB.'
                }


    except Error as e:
        # En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e}"
        }

    finally:
        cerrar_conexion()
        print("Conexión a MySQL cerrada")
```

```
def getImeiImport(imei):
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT IMEI1,IMEI2 FROM TERM_IMPORT WHERE IMEI1 =
%s OR IMEI2 = %s", (imei, imei))
        resultado = cursor.fetchone()
        print(resultado)

        if resultado:
            # Eliminar las claves cuyo valor sea None
            resultado_filtrado = {key: value for key, value in
resultado.items() if value is not None}
            print(f"Resultado filtrado: {resultado_filtrado}")
            return resultado_filtrado
        else:
            print("No se encontró el IMEI")
            return {}

        #return resultado

    except Exception as e:
        print(f"Error al verificar IMEI: {e}")
        return None

def deleteListaBlanca(imei):
    #DECLARAR EL BORRADO DE LB DE UN IMEI.
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        print('ENTRO PARA BORRAR')

        imeiEnImp = getImeiImport(imei)
        print(imeiEnImp.get('IMEI1'),imeiEnImp.get('IMEI2'))
        cantidad = len(imeiEnImp) if imeiEnImp else 0
        cursor = connection.cursor()

        if cantidad == 1:
            cursor.execute("DELETE FROM LISTA_BLANCA WHERE IMEI = %s",
(imeiEnImp.get('IMEI1'),))
            print('BORRÉ en lista blanca')

            #LLAMAR A INSERTAR A LBH.

        elif cantidad == 2:
            cursor.execute("DELETE FROM LISTA_BLANCA WHERE IMEI = %s",
(imeiEnImp.get('IMEI1'),))
```

```
            cursor.execute("DELETE FROM LISTA_BLANCA WHERE IMEI = %s",
(imeiEnImp.get('IMEI2'),))
            print('BORRÉ en lista blanca 2 IMEI')
        else:
            print('No se encontró IMEI en EMP_IMPORT')
        connection.commit()
    except Exception as e:
        print(f"Error al ejecutar la función deleteListaBlanca: {e}")
        return None
def deleteListaBlancaHist(imei):
    #DECLARAR EL BORRADO DE LB DE UN IMEI.
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        print('ENTRE PARA BORRAR DE LA LISTA BLANCA HIST')

        imeiEnImp = getImeiImport(imei)
        print(imeiEnImp.get('IMEI1'),imeiEnImp.get('IMEI2'))
        cantidad = len(imeiEnImp) if imeiEnImp else 0
        cursor = connection.cursor()

        if cantidad == 1:
            cursor.execute("DELETE FROM LISTA_BLANCA_HIST WHERE IMEI =
%s", (imeiEnImp.get('IMEI1'),))
            print('BORRÉ en lista blancaHIST 1 IMEI')

            #LLAMAR A INSERTAR A LBH.


        elif cantidad == 2:
            cursor.execute("DELETE FROM LISTA_BLANCA_HIST WHERE IMEI =
%s", (imeiEnImp.get('IMEI1'),))
            cursor.execute("DELETE FROM LISTA_BLANCA_HIST WHERE IMEI =
%s", (imeiEnImp.get('IMEI2'),))
            print('BORRÉ en lista blancaHIST 2 IMEI.')
        else:
            print('No se encontró IMEI en EMP_IMPORT')
        connection.commit()
    except Exception as e:
        print(f"Error al ejecutar la función deleteListaBlanca: {e}")
        return None

def deleteListaGris(imei):
    #DECLARAR EL BORRADO DE LG DE UN IMEI.
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        print('ENTRE PARA BORRAR DE LA LISTA GRIS')
```

```
        imeiEnImp = getImeiImport(imei)
        print(imeiEnImp.get('IMEI1'),imeiEnImp.get('IMEI2'))
        cantidad = len(imeiEnImp) if imeiEnImp else 0
        cursor = connection.cursor()

        if cantidad == 1:
            cursor.execute("DELETE FROM LISTA_GRIS WHERE IMEI = %s",
(imeiEnImp.get('IMEI1'),))
            print('BORRÉ en lista GRIS 1 IMEI')

            #LLAMAR A INSERTAR A LBH.

        elif cantidad == 2:
            cursor.execute("DELETE FROM LISTA_GRIS WHERE IMEI = %s",
(imeiEnImp.get('IMEI1'),))
            cursor.execute("DELETE FROM LISTA_GRIS WHERE IMEI = %s",
(imeiEnImp.get('IMEI2'),))
            print('BORRÉ en lista GRIS 2 IMEI')
        else:
            print('No se encontró IMEI en EMP_IMPORT')
        connection.commit()
    except Exception as e:
        print(f"Error al ejecutar la función deleteListaBlanca: {e}")
        return None

def insertListaGris(imei,idmsg,numeroServicio,imsi,motivoReporte):
    try:

        if connection is None or not connection.is_connected():
            conectar_db()
        if idmsg is None:
            raise Exception("No se pudo obtener el IDMSG generado")
        cursor = connection.cursor()
        # Generar un IDPROC
        idproc = ''.join([str(random.randint(0, 9)) for _ in range(10)])


        cantidadLB = nTerminalesImp(imei)
        if cantidadLB == 2:
            print('2IMEIS en LB')

            cursor = connection.cursor(dictionary=True)
            cursor.execute("SELECT IMEI1,IMEI2 FROM TERM_IMPORT WHERE
imei1 = %s OR imei2 = %s", (imei,imei,))
            result = cursor.fetchone()
            print(result)

            cursor1 = connection.cursor()
```

```
            query2 = """
INSERT INTO LISTA_GRIS (IMEI, IDMSG, IDPROC, MSISND, IMSI,
MOTIVO_REPORTE, FECHA_REGISTRO, ROBADO, NO_BLANCA, PROCESO_ORIGEN)
VALUES (%s, %s, %s, %s, %s, %s, CURRENT_TIMESTAMP, %s, %s, %s)
            """

##VALIDAR SI ESTA EN LA BLANCA CON VALORES 1 o =, también validar si es S
o P
            params1 = (result.get('IMEI1'), idmsg, idproc,
numeroServicio, imsi, motivoReporte,'1', '0', 'PROC_SPR'  # ROBADO = '1'
y PROCESO_ORIGEN = 'PROC_SPR')
            cursor1.execute(query2, params1)
            params2 = (result.get('IMEI2'), idmsg, idproc,
numeroServicio, imsi, motivoReporte, '1', '0', 'PROC_SPR'  # ROBADO = '1'
y PROCESO_ORIGEN = 'PROC_SPR')
            cursor1.execute(query2, params2)
# Si ambas inserciones fueron exitosas, realizar commit de la transacción
            connection.commit()
            print('INSERTÉ EN GRIS 2 IMEI')



        elif cantidadLB == 1:
            print('1 IMEIS en LB')

            cursor = connection.cursor(dictionary=True)
            cursor.execute("SELECT IMEI1 FROM TERM_IMPORT WHERE imei1 =
%s OR imei2 = %s", (imei,imei,))
            result = cursor.fetchone()
            print(result)

            cursor1 = connection.cursor()

            query2 = """
INSERT INTO LISTA_GRIS (IMEI, IDMSG, IDPROC, MSISND, IMSI,
MOTIVO_REPORTE, FECHA_REGISTRO, ROBADO, NO_BLANCA, PROCESO_ORIGEN
)
VALUES (%s, %s, %s, %s, %s, %s, CURRENT_TIMESTAMP, %s, %s, %s)
            """
##VALIDAR SI ESTA EN LA BLANCA CON VALORES 1 o =, también validar si es S
o P
            params1 = (
                result.get('IMEI1'), idmsg, idproc, numeroServicio, imsi,
motivoReporte, '1', '0', 'PROC_SPR'  # ROBADO = '1' y PROCESO_ORIGEN =
'PROC_SPR')
            cursor1.execute(query2, params1)

# Si ambas inserciones fueron exitosas, realizar commit de la transacción
            connection.commit()
```

```
            print('INSERTÉ EN GRIS 1 IMEI')


        else:
            print('NO EXISTEEE')
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'cantidad':cantidadLB,
            'codigoRechazo':'No existe en la LB.'
            }

        return {
            'statusCode': 200,
            'message': 'Registro agregado exitosamente'
        }

    except Error as e:
        # En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e}"
        }

    finally:
        cerrar_conexion()
        print("Conexión a MySQL cerrada")

def deleteListaNegra(imei):
#DECLARAR EL BORRADO DE LN DE UN IMEI.
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        print('ENTRE PARA BORRAR DE LA LISTA NEGRA')

        imeiEnImp = getImeiImport(imei)
        print(imeiEnImp.get('IMEI1'),imeiEnImp.get('IMEI2'))
        cantidad = len(imeiEnImp) if imeiEnImp else 0
        cursor = connection.cursor()

        if cantidad == 1:
            cursor.execute("DELETE FROM LISTA_NEGRA WHERE IMEI = %s",
(imeiEnImp.get('IMEI1'),))
            print('BORRÉ en lista NEGRA 1 IMEI')


#LLAMAR A INSERTAR A LBH.
        elif cantidad == 2:
            cursor.execute("DELETE FROM LISTA_NEGRA WHERE IMEI = %s",
(imeiEnImp.get('IMEI1'),))
```

```
            cursor.execute("DELETE FROM LISTA_NEGRA WHERE IMEI = %s",
(imeiEnImp.get('IMEI2'),))
            print('BORRÉ en lista NEGRA 2 IMEI')
        else:
            print('No se encontró IMEI en EMP_IMPORT')
        connection.commit()
    except Exception as e:
        print(f"Error al ejecutar la función deleteListaNegra: {e}")
        return None


def insertListaBlancaHist(imei,idmsg,motivoReporte):
    '''
Insertar en ListaGris cuando un IMEI está en la Lista Blanca.
Args:
imei (str): El IMEI a insertar. Debe ser una cadena de 15 dígitos
numéricos.
Returns:
bool: Retorna `True` si el IMEI está en la ListaNegra,
`False` si no lo está.
'''
    cantidadLB = nTerminalesImp(imei)
    if cantidadLB == 2:
        print('2IMEIS en LB2')
        #insertListaGrisLB(imei,idmsg,motivoReporte)
        #LLAMAR A LA FUNCION PARA SACAR 2IMEI de la LB

        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT IMEI1,IMEI2 FROM TERM_IMPORT WHERE imei1 =
%s OR imei2 = %s", (imei,imei,))
        result = cursor.fetchone()

        cursor1 = connection.cursor()
        query = """
            INSERT INTO LISTA_BLANCA_HIST (IMEI, IDMSG, IDREGISTRO,
MOTIVO_REPORTE, FECHA_REGISTRO, PROCESO_ORIGEN)
            VALUES (%s, %s, NULL, %s, CURRENT_TIMESTAMP, %s)
        """
        lb1 = (
            result.get('IMEI1'), idmsg, motivoReporte, 'PROC_SPR'
        )
        cursor1.execute(query, lb1)
        lb2 = (
            result.get('IMEI2'), idmsg, motivoReporte, 'PROC_SPR'
        )
        cursor1.execute(query, lb2)
        connection.commit()
        print("Registro insertado exitosamente en la LB HIST")
```

```
        mensaje = {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'codigoRechazo':'Tiene 2 IMEIS'
        }
        return mensaje


    elif cantidadLB == 1:
        print('1 IMEIS en LB1')
        #insertListaGrisLB(imei,idmsg,motivoReporte)
        #LLAMAR A LA FUNCION PARA SACAR 2IMEI de la LB

        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT IMEI1 FROM TERM_IMPORT WHERE imei1 = %s OR
imei2 = %s", (imei,imei,))
        result = cursor.fetchone()

        cursor1 = connection.cursor()
        query = """
INSERT INTO LISTA_BLANCA_HIST (IMEI, IDMSG, IDREGISTRO, MOTIVO_REPORTE,
FECHA_REGISTRO, PROCESO_ORIGEN)
VALUES (%s, %s, NULL, %s, CURRENT_TIMESTAMP, %s)
        """
        lb1 = (
            result.get('IMEI1'), idmsg, motivoReporte, 'PROC_SPR'
        )
        cursor1.execute(query, lb1)
        connection.commit()
        print("Registro insertado exitosamente en la LB HIST")
        return {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'codigoRechazo':'Tiene 1 IMEI'
        }

    else:
        print('NO EXISTE')
        return {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'cantidad':cantidadLB,
        'codigoRechazo':'No existe en la LB.'
        }


def insertListaBlanca(imei,idmsg,motivoReporte):
    '''
    Insertar en Lista Gris cuando un IMEI está en la Lista Blanca.

    Args:
```

```
        imei (str): El IMEI a insertar. Debe ser una cadena de 15 dígitos
numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    cantidadLB = nTerminalesImp(imei)
    if cantidadLB == 2:
        print('2IMEIS en LB')
        #insertListaGrisLB(imei,idmsg,motivoReporte)
        #LLAMAR A LA FUNCION PARA SACAR 2IMEI de la LB

        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT IMEI1,IMEI2 FROM TERM_IMPORT WHERE imei1 =
%s OR imei2 = %s", (imei,imei,))
        result = cursor.fetchone()

        cursor1 = connection.cursor()
        query = """
            INSERT INTO LISTA_BLANCA (IMEI, IDMSG, IDREGISTRO,
MOTIVO_REPORTE, FECHA_REGISTRO, PROCESO_ORIGEN)
            VALUES (%s, %s, NULL, %s, CURRENT_TIMESTAMP, %s)
        """
        lb1 = (
            result.get('IMEI1'), idmsg, motivoReporte, 'PROC_SPR'
        )
        cursor1.execute(query, lb1)
        lb2 = (
            result.get('IMEI2'), idmsg, motivoReporte, 'PROC_SPR'
        )
        cursor1.execute(query, lb2)
        connection.commit()
        print("Registro insertado exitosamente en la LB HIST")


        mensaje = {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'codigoRechazo':'Tiene 2 IMEIS'
        }
        return mensaje


    elif cantidadLB == 1:
        print('1 IMEIS en LB1')
        #insertListaGrisLB(imei,idmsg,motivoReporte)
        #LLAMAR A LA FUNCION PARA SACAR 2IMEI de la LB

        cursor = connection.cursor(dictionary=True)
```

```
        cursor.execute("SELECT IMEI1 FROM TERM_IMPORT WHERE imei1 = %s OR
imei2 = %s", (imei,imei,))
        result = cursor.fetchone()

        cursor1 = connection.cursor()
        query = """
INSERT INTO LISTA_BLANCA (IMEI, IDMSG, IDREGISTRO, MOTIVO_REPORTE,
FECHA_REGISTRO, PROCESO_ORIGEN)
VALUES (%s, %s, NULL, %s, CURRENT_TIMESTAMP, %s)
        """
        lb1 = (
            result.get('IMEI1'), idmsg, motivoReporte, 'PROC_SPR'
        )
        cursor1.execute(query, lb1)
        connection.commit()
        print("Registro insertado exitosamente en la LB")
        return {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'codigoRechazo':'Tiene 1 IMEI'
        }

    else:
        print('NO EXISTEEE')
        return {
        'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
        'cantidad':cantidadLB,
        'codigoRechazo':'No existe en la LB.'
        }
def motivoReporteLN(imei):
    '''
Valida el motivo de reporte de un IMEI en LN.
Args:
imei (str): El IMEI a verificar. Debe ser una cadena de 15 dígitos
numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT 1 FROM LISTA_NEGRA WHERE imei = %s AND
(NO_BLANCA='1' OR INVALIDO = '1' OR DUPLICADO='1')", (imei,))

        resultado = cursor.fetchone()
        print(f'VALIDANDO SI TIENE OTRO MOTIVO EN NEGRA: {resultado}')
        return resultado is not None
```

```
    except Exception as e:
        print(f"Error al verificar la lista negra: {e}")
        return False
def motivoReporteLG(imei):
    '''
    Valida el motivo de reporte de un IMEI en LG.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT 1 FROM LISTA_GRIS WHERE imei = %s AND
NO_BLANCA='1'", (imei,))

        resultado = cursor.fetchone()
        print(f'VALIDANDO SI TIENE OTRO MOTIVO EN GRIS: {resultado}')
        return resultado is not None

    except Exception as e:
        print(f"Error al verificar la lista gris: {e}")
        return False

def sacarDatos(imei):
    try:
        if connection is None or not connection.is_connected():
            conectar_db()

        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT * FROM MSG_ROBO_RECUP WHERE imei = %s",
(imei,))
        result = cursor.fetchone()
        #print(result)
        if result:
            msisnd = result.get('MSISND')
            imsi = result.get('IMSI')
            concesionario = result.get('CONCESIONARIO')
            motivo_reporte = result.get('MOTIVO_REPORTE')
        else:
            msisnd = imsi= concesionario= motivo_reporte = None
```

```
    return {
        'statusCode': 200,
        'imei':imei,
        'msisnd':msisnd,
        'imsi':imsi,
        'concesionario':concesionario,
        'motivo_reporte':motivo_reporte
    }

except Error as e:
    # En caso de error, devuelve el mensaje de error y código 500
    return {
        'statusCode': 500,
        'body': f"Error: {e}"
    }

finally:
    cerrar_conexion()
    print("Conexión a MySQL cerrada")
```

II) API Gestión de acceso a la red

```
import mysql.connector
import misFunciones as mf
from datetime import datetime
from mysql.connector import Error
import boto3
import json
import random
import os

tac_invalidos = ['00000000', '11111111', '12345678']

def lambda_handler(event, context):
    global connection

    concesionario,imei,numeroServicio,imsi =
event.get('concesionario'),event.get('imei'),event.get('numeroServicio'),
event.get('imsi')
    tac = imei[:8]

    if not imei:
        codigoError = 'ERRACCESS000001'
        mf.insertarConsultaCEIR(concesionario, imei, numeroServicio,
imsi,codigoError)

        return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'codigoRechazo': codigoError,
            'mensaje': 'Falta o esta vacío el campo IMEI en la solicitud'
        }
    elif len(imei) != 15 or not imei.isdigit():
        codigoError = 'ERRACCESS000002'
        mf.insertarConsultaCEIR(concesionario, imei, numeroServicio,
imsi,codigoError)
        return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'codigoRechazo': codigoError,
            'mensaje':'IMEI - longitud incorrecta o valor inválido.'
        }
    elif tac in tac_invalidos:
        codigoError = 'ERRACCESS00003'
        mf.insertarConsultaCEIR(concesionario, imei, numeroServicio,
imsi,codigoError)
        return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'codigoRechazo': codigoError,
            'mensaje':'TAC es inválido.'
```

```
        }
    try:
        #Comprobar si está en LN
        if mf.inListaNegra(imei) == True:
            #Soltar mensaje de "Acceso bloqueado."

            mf.insertarConsultaCEIR(concesionario, imei, numeroServicio,
imsi)
            mensajeJson='BLOCKED'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado': 'OK',
            'imei': imei,
            'imsi': imsi,
            'status':mensajeJson

        elif mf.inListaBlanca(imei) == True:
      mf.insertarConsultaCEIR(concesionario, imei, numeroServicio, imsi)
            mensajeJson='PERMITTED'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado': 'OK',
            'imei': imei,
            'imsi': imsi,
            'mensaje':mensajeJson
            }

        elif mf.inListaGris(imei) == True and mf.inListaNegra(imei) ==
False:
            print('Entré al ELIF Está en GRIS pero no en NEGRA')
            if mf.isRobadoLG(imei) == 1:
                lastDigitImei = imei[-4:]
mensaje = f'Está haciendo uso de un terminal móvil con IMEI
****{lastDigitImei}, reportado como ROBADO debe entregar este equipo en
la comisaría más cercana, y solicitar se entregue la constancia de
devolución, en caso no se realice se suspenderá su servicio en 30 días.'
                mf.sendSMS(mensaje)
mf.insertarConsultaCEIR(concesionario, imei, numeroServicio, imsi)
                mensajeJson='ALLOWED'
                return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'resultado': 'OK',
                'imei': imei,
                'imsi': imsi,
                'codigoRechazo':mensajeJson
                }

            if mf.isNoBlancaLG(imei) == 1:
                lastDigitImei = imei[-4:]
```

```
mensaje = f'Está haciendo uso de un terminal móvil con IMEI
****{lastDigitImei}, no registrado en Lista Blanca, si está en proceso de
compra, no continúe y devuelva el equipo al vendedor.'
                mf.sendSMS(mensaje)
                #print(mensaje)
mf.insertarConsultaCEIR(concesionario, imei, numeroServicio, imsi)
                mensajeJson='ALLOWED'
                return {
                'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
                'resultado': 'OK',
                'imei': imei,
                'imsi': imsi,
                'codigoRechazo':mensajeJson}
        else:
lastDigitImei = imei[-4:]
mensaje = f'Está haciendo uso del terminal móvil con IMEI
****{lastDigitImei}, el cual no está registrado, en los siguientes días
no tendrá acceso a la red.'

            #print(mensaje)
mf.insertarConsultaCEIR(concesionario, imei, numeroServicio, imsi)
            mf.insertListaGris(imei,numeroServicio,imsi)
            print("PASE GRIS")
            mf.sendSMS(mensaje)
            mensajeJson='ALLOWED'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado': 'OK',
            'imei': imei,
            'imsi': imsi,
            'codigoRechazo':mensajeJson
            }


    except Error as e:
        # En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e} EEEEEEEE"
        }

    finally:
        # Cerrar la conexión si está abierta
        mf.cerrar_conexion()
        print("Conexión a MySQL cerrada")
```

Función de registro de información en la base de datos: MisFunciones.py, para el API Gestión de acceso a la red

```
import mysql.connector
from datetime import datetime
from mysql.connector import Error
import json
import boto3
import random
import os

connection = None
def conectar_db():
    global connection
    try:
        connection = mysql.connector.connect(
            host= os.getenv('DATABASE_HOST'),
            database= os.getenv('DATABASE_PRO'),
            user= os.getenv('DATABASE_USER'),
            password= os.getenv('DATABASE_PASS')
        )
        if connection.is_connected():
            print("Conexión exitosa a la base de datos.")
    except mysql.connector.Error as err:
        print(f"Error al conectar a la base de datos: {err}")

def cerrar_conexion():
    global connection
    if connection and connection.is_connected():
        connection.close()
        print("Conexión cerrada.")

def inListaNegra(imei):
    '''
    Valida si el IMEI existe en la Lista Negra.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_NEGRA WHERE imei = %s",
(imei,))
```

```
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def inListaGris(imei):
    '''
    Valida si el IMEI existe en la Lista Gris.
    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaGris,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_GRIS WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def inListaBlanca(imei):
    '''
    Valida si el IMEI existe en la Lista Blanca.
    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaBlanca,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_BLANCA WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
```

```
        return True
    else:
        return False

def insertarConsultaCEIR(concesionario, imei, numeroServicio,imsi,
codigoRechazo=None):
    try:
        if connection is None or not connection.is_connected():
            conectar_db()


        cursor = connection.cursor()
        query = """
INSERT INTO CONSULTAS_CEIR (ID_CONSULTA, CONCESIONARIO, IMEI, MSISND,
IMSI, FECHA_REGISTRO, COD_CODIGO_RECHAZO)
            VALUES (
                NULL, %s, %s, %s, %s, CURRENT_TIMESTAMP, %s)
        """
        params = (
            concesionario, imei, numeroServicio, imsi, codigoRechazo if
codigoRechazo is not None else None
        )
        cursor.execute(query, params)
        connection.commit()
        print("Insertado en CEIR")


    except Error as e:
# En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e}"
        }

    finally:
        cerrar_conexion()
        print("Conexión a MySQL cerrada")

def isRobadoLG(imei):

    '''
Valida el motivo de reporte de un IMEI en LN.
    Args:
      imei (str): El IMEI a verificar. Debe ser una cadena de 15 dígitos
numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
```

```
    '''
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT 1 FROM LISTA_GRIS WHERE imei = %s AND
ROBADO='1'", (imei,))
        resultado = cursor.fetchone()
        return resultado is not None

    except Exception as e:
        print(f"Error al verificar la lista negra: {e}")
        return False


def isNoBlancaLG(imei):
    '''
    Valida el motivo de reporte de un IMEI en LG.
    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.
    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor(dictionary=True)
        cursor.execute("SELECT 1 FROM LISTA_GRIS WHERE imei = %s AND
NO_BLANCA='1'", (imei,))

        resultado = cursor.fetchone()
        return resultado is not None

    except Exception as e:
        print(f"Error al verificar la lista gris: {e}")
        return False

def sendSMS(mensaje):
    sns_client = boto3.client('sns', region_name='us-east-2')
    # Solo números verificados.
    phone_number = '+51979962812'#'+51979962812'
    #message = f'¡Hola! Este es un mensaje enviado desde AWS SNS a través
de Lambda. TAC de la consulta.'
    #print(mensaje)
    try:
        # Enviar el SMS a través de SNS
        response = sns_client.publish(
```

```
            PhoneNumber=phone_number,
            Message=mensaje
        )

        # Regresar la respuesta de SNS
        return {
            'statusCode': 200,
            'body': json.dumps(f'SMS enviado exitosamente. Response:
{response}')
        }

    except Exception as e:
        # Manejo de errores
        return {
            'statusCode': 500,
            'body': json.dumps(f'Ocurrió un error al enviar el SMS:
{str(e)}')
        }

def nTerminalesImp(imei):
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor()
        query = """
            SELECT
                CASE
WHEN COUNT(DISTINCT IMEI1) + COUNT(DISTINCT IMEI2) = 2 THEN 2
WHEN COUNT(DISTINCT IMEI1) + COUNT(DISTINCT IMEI2) = 1 THEN 1
ELSE 0
                END AS resultado
            FROM
                TERM_IMPORT
            WHERE
                IMEI1 = %s OR IMEI2 = %s;
            """
        cursor.execute(query, (imei, imei))
        resultado = cursor.fetchone()
        if resultado:
            return resultado[0]  # Devolver el resultado (puede ser 2 ó
1)
        else:
            return 0
    except Exception as e:
        print(f"Error al verificar IMEI: {e}")
        return None
```

```
def insertListaGris(imei,numeroServicio,imsi):
    try:

        if connection is None or not connection.is_connected():
            conectar_db()

        cursor = connection.cursor()
        # Generar un IDPROC
        idproc = ''.join([str(random.randint(0, 9)) for _ in range(10)])


        cantidadLB = nTerminalesImp(imei)
        if cantidadLB == 2:
            print('2IMEIS en LB')

            cursor = connection.cursor(dictionary=True)
            cursor.execute("SELECT IMEI1,IMEI2 FROM TERM_IMPORT WHERE
imei1 = %s OR imei2 = %s", (imei,imei,))
            result = cursor.fetchone()
            print(result)

            cursor1 = connection.cursor()

            query2 = """
INSERT INTO LISTA_GRIS (IMEI, IDMSG, IDPROC, MSISND, IMSI,
MOTIVO_REPORTE, FECHA_REGISTRO, ROBADO, NO_BLANCA, PROCESO_ORIGEN)
VALUES (%s, NULL, %s, %s, %s, NULL, CURRENT_TIMESTAMP, %s, %s, %s)
            """
#VALIDAR SI ESTA EN LA BLANCA CON VALORES 1 o =, también validar si es S
o P
            params1 = (
                result.get('IMEI1'), idproc, numeroServicio, imsi,
                '1', '0', 'PROC_REDACCESS'  # ROBADO = '1' y
PROCESO_ORIGEN = 'PROC_REDACCESS'
            )
            cursor1.execute(query2, params1)
            params2 = (
result.get('IMEI2'), idproc, numeroServicio, imsi, '1', '0',
'PROC_REDACCESS'
# ROBADO = '1' y PROCESO_ORIGEN = 'PROC_REDACCESS')
            cursor1.execute(query2, params2)
# Si ambas inserciones fueron exitosas, realizar commit de la transacción
            connection.commit()
            print('INSERTÉ EN GRIS 2 IMEI')



        elif cantidadLB == 1:
            print('1 IMEIS en LB')
```

```
            cursor = connection.cursor(dictionary=True)
            cursor.execute("SELECT IMEI1 FROM TERM_IMPORT WHERE imei1 =
%s OR imei2 = %s", (imei,imei,))
            result = cursor.fetchone()
            print(result)

            cursor1 = connection.cursor()

            query2 = """
INSERT INTO LISTA_GRIS (IMEI, IDMSG, IDPROC, MSISND, IMSI,
MOTIVO_REPORTE, FECHA_REGISTRO, ROBADO, NO_BLANCA, PROCESO_ORIGEN)
VALUES (%s, NULL, %s, %s, %s, NULL, CURRENT_TIMESTAMP, %s, %s, %s)
            """
#VALIDAR SI ESTA EN LA BLANCA CON VALORES 1 o =, también validar si es S
o P
            params1 = (
                result.get('IMEI1'), idproc, numeroServicio, imsi,
                '1', '0', 'PROC_REDACCESS'  # ROBADO = '1' y
PROCESO_ORIGEN = 'PROC_REDACCESS'
            )
            cursor1.execute(query2, params1)

            # Si ambas inserciones fueron exitosas, realizar commit de la
transacción
            connection.commit()
            print('INSERTÉ EN GRIS 1 IMEI')


        else:
            print('NO EXISTE')


            cursor1 = connection.cursor()

            query2 = """
INSERT INTO LISTA_GRIS (
IMEI, IDMSG, IDPROC, MSISND, IMSI, MOTIVO_REPORTE, FECHA_REGISTRO,
ROBADO, NO_BLANCA, PROCESO_ORIGEN)
                VALUES (%s, NULL, %s, %s, %s, NULL, CURRENT_TIMESTAMP,
%s, %s, %s)
            """
#VALIDAR SI ESTA EN BLANCA CON VALORES 1 o =, también validar si es S o P
            params1 = (
                imei, idproc, numeroServicio, imsi,
                '1', '0', 'PROC_REDACCESS'  # ROBADO = '1' y
PROCESO_ORIGEN = 'PROC_REDACCESS'
            )
            cursor1.execute(query2, params1)
```

```
# Si ambas inserciones fueron exitosas, realizar commit de la transacción
            connection.commit()
            print('INSERTÉ EN GRIS IMEI QUE SOLICITÓ ACCESO A LA RED.')


        return {
            'statusCode': 200,
            'message': 'Registro agregado exitosamente'
        }

    except Error as e:
# En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e}"
        }

    finally:
        cerrar_conexion()
        print("Conexión a MySQL cerrada")
```

III) API Registro Lista Blanca

```
import mysql.connector
import misFunciones as mf
from datetime import datetime
from mysql.connector import Error
import boto3
import json
import random
import os

tac_invalidos = ['00000000', '11111111', '12345678']

def lambda_handler(event, context):
    global connection
    '''
    Proceso de reporte SPR
    '''
    idMsg,imei,fechaRegistro,procesoOrigen =
event.get('idMsg'),event.get('imei'),event.get('fechaRegistro'),event.get
('procesoOrigen')
    tac = imei[:8]
    #print(mf.mostrarListaGris())
    if not imei:

        return {
            'statusCode': 400,
            'error': 'Falta o está vacío el campo IMEI en la solicitud'
        }
    elif len(imei) != 15 or not imei.isdigit():

        return {
            'statusCode': 400,
            'codigoRechazo':'RNTSG000002',
            'descripcion':'IMEI - longitud incorrecta o valor inválido.'
        }
    elif tac in tac_invalidos:
        return {
            'statusCode': 400,
            'codigoRechazo':'TAC es inválido.'
        }

    try:
        #Comprobar si está en LN - CASO DE USO A-1
        if mf.inListaNegra(imei) == True:
            '''
```

```
            Rechazo por estar en LN
            '''
            codigoError = 'APIREGLB001'
            mensaje='IMEI registrado en Lista Negra.'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado': 'Error',
            'codigoRechazo': codigoError,
            'mensaje': mensaje
            }
        #CASO DE USO A-2.
        elif mf.inListaBlanca(imei) == False and mf.inListaGris(imei) ==
True and mf.obtenerMotivoReporte(imei) == 'Robado':
            codigoError = 'APIREGLB002'
            mensaje='IMEI registrado en la Lista Gris por Robado.'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado': 'Error',
            'codigoRechazo': codigoError,
            'mensaje': mensaje
            }
        #CASO DE USO A-3.
        elif mf.inListaBlanca(imei) == False and mf.inListaGris(imei) ==
True and mf.obtenerMotivoReporte(imei) == 'NoBlanca':
            mf.deleteListaGris(imei)
            mf.insertListaBlanca(imei,idMsg,procesoOrigen)
            resultado='Regularizado'
            mensaje = 'Se regularizó el registro del IMEI.'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado': resultado,
            'mensaje' : mensaje
            }
        #CASO DE USO A-4.
        elif mf.inListaBlanca(imei) == True and
mf.inListaNegra(imei)==False and mf.inListaGris(imei)==False:
            codigoError = 'APIREGLB003'
            mensaje='IMEI previamente registrado en Lista Blanca.'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado': 'Error',
            'codigoRechazo': codigoError,
            'mensaje': mensaje
            }
        #CASO DE USO A-5
        elif mf.inListaBlanca(imei) == False and
mf.inListaNegra(imei)==False and mf.inListaGris(imei)==False:
            mf.insertListaBlanca(imei,idMsg,procesoOrigen)
            resultado='Registrado'
```

```
            mensaje = 'IMEI se registró en la Lista Blanca.'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado':resultado,
            'mensaje' : mensaje
            }
        #POR SI NO SE CUMPLEN LOS CASOS DE USO
        else:
            resultado='Fallo en los casos de uso'
            mensaje = 'El IMEI no cumple con ninguno de los casos.'
            return {
            'fechaMensaje': datetime.now().strftime('%Y%m%d%H%M%S'),
            'resultado':resultado,
            'mensaje' : mensaje
            }

    except Error as e:
        # En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e}"
        }

    finally:
        # Cerrar la conexión si está abierta
        mf.cerrar_conexion()
        print("Conexión a MySQL cerrada")
```

Función de registro de información en la base de datos: MisFunciones.py para API Registro Lista Blanca

```
import mysql.connector
from datetime import datetime
from mysql.connector import Error
import json
import random
import os

connection = None
def conectar_db():
    global connection
    try:
        connection = mysql.connector.connect(
            host= os.getenv('DATABASE_HOST'),
            database= os.getenv('DATABASE_PRO'),
            user= os.getenv('DATABASE_USER'),
```

```
            password= os.getenv('DATABASE_PASS')
        )
        if connection.is_connected():
            print("Conexión exitosa a la base de datos.")
    except mysql.connector.Error as err:
        print(f"Error al conectar a la base de datos: {err}")

def cerrar_conexion():
    global connection
    if connection and connection.is_connected():
        connection.close()
        print("Conexión cerrada.")

def inListaNegra(imei):
    '''
    Valida si el IMEI existe en la Lista Negra.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaNegra,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_NEGRA WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def inListaGris(imei):
    '''
    Valida si el IMEI existe en la Lista Gris.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        bool: Retorna `True` si el IMEI está en la ListaGris,
              `False` si no lo está.
    '''
```

```
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_GRIS WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def inListaBlanca(imei):
    '''
    Valida si el IMEI existe en la Lista Blanca.
    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.
    Returns:
        bool: Retorna `True` si el IMEI está en la ListaBlanca,
              `False` si no lo está.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT COUNT(1) FROM LISTA_BLANCA WHERE imei = %s",
(imei,))
    result = cursor.fetchone()
    count = result[0]
    if count > 0:
        return True
    else:
        return False

def insertarConsultaCEIR(concesionario, imei, numeroServicio,imsi,
codigoRechazo=None):
    try:
        if connection is None or not connection.is_connected():
            conectar_db()


        cursor = connection.cursor()
        query = """
INSERT INTO CONSULTAS_CEIR (ID_CONSULTA, CONCESIONARIO, IMEI, MSISND,
IMSI, FECHA_REGISTRO, COD_CODIGO_RECHAZO)
VALUES (NULL, %s, %s, %s, %s, CURRENT_TIMESTAMP, %s)
        """
        params = (
```

```
            concesionario, imei, numeroServicio, imsi, codigoRechazo if
codigoRechazo is not None else None
        )

        cursor.execute(query, params)
        connection.commit()
        print("Insertado en CEIR")


    except Error as e:
        # En caso de error, devuelve el mensaje de error y código 500
        return {
            'statusCode': 500,
            'body': f"Error: {e}"
        }

    finally:
        cerrar_conexion()
        print("Conexión a MySQL cerrada")

def obtenerMotivoReporte(imei):
    '''
    Valida el motivo de reporte del IMEI en la Lista Gris.

    Args:
        imei (str): El IMEI a verificar. Debe ser una cadena de 15
dígitos numéricos.

    Returns:
        str: Retorna 'Robado' si el IMEI está reportado como robado,
             'NoBlanca' si está reportado como no blanca,
             'Ambos' si está reportado por ambas razones,
             'Ninguno' si no está reportado.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT ROBADO, NO_BLANCA FROM LISTA_GRIS WHERE imei =
%s", (imei,))
    result = cursor.fetchone()
    if result:
        robado, no_blanca = result
        if robado == '1' and no_blanca == '1':
            return 'Ambos'
        elif robado == '1':
            return 'Robado'
        elif no_blanca == '1':
            return 'NoBlanca'
    return 'Ninguno'
```

```
def insertListaBlanca(imei,idmsg,procesoOrigen):
    '''
    Insertar en Lista Blanca el IMEI brindado

    Args:

        imei (str): El IMEI a insertar. Debe ser una cadena de 15 dígitos
numéricos.
        idmsg: idMsg obtenido del json enviado.
        procesoOrigen: proveniencia del imei.
    '''
    cursor = connection.cursor()
    query = """
        INSERT INTO LISTA_BLANCA (
            IMEI, IDMSG, IDREGISTRO, MOTIVO_REPORTE, FECHA_REGISTRO,
PROCESO_ORIGEN
        )
        VALUES (
            %s, %s, NULL, NULL, CURRENT_TIMESTAMP, %s
        )
    """
    lb1 = (
        imei, idmsg, procesoOrigen
    )
    cursor.execute(query, lb1)
    connection.commit()
    print("Registro insertado exitosamente en la LB")

def deleteListaGris(imei):
    #DECLARAR EL BORRADO DE LG DE UN IMEI.
    try:
        if connection is None or not connection.is_connected():
            conectar_db()
        cursor = connection.cursor()
        cursor.execute("DELETE FROM LISTA_GRIS WHERE IMEI = %s", (imei,))
        connection.commit()
    except Exception as e:
        print(f"Error al ejecutar la función deleteListaGris: {e}")
        return None

def mostrarListaGris():
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM LISTA_GRIS")
    result = cursor.fetchall()
    return result
```

```
def mostrarListaNegra():
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM LISTA_NEGRA")
    result = cursor.fetchall()
    formatted_results = []
    for row in result:
        formatted_row = f"IMEI: {row[0]}, ID: {row[1]}, Número: {row[2]},
Otro Número: {row[3]}, Código: {row[4]}, Género: {row[5]}, Fecha:
{row[6]}, Columna1: {row[7]}, Columna2: {row[8]}, Columna3: {row[9]},
Columna4: {row[10]}, Proceso: {row[11]}"
        formatted_results.append(formatted_row)

    return "\n".join(formatted_results)

def mostrarNombresColumnas():
    '''
    Obtiene y muestra los nombres de las columnas de la tabla
LISTA_NEGRA.

    Returns:
        list: Una lista con los nombres de las columnas.
    '''
    if connection is None or not connection.is_connected():
        conectar_db()
    cursor = connection.cursor()
    cursor.execute("SELECT * FROM LISTA_NEGRA LIMIT 1")
    column_names = [desc[0] for desc in cursor.description]
    return column_names
```