ESCUELA POLITÉCNICA NACIONAL

DEPARTAMENTO DE ELECTRÓNICA, TELECOMUNICACIONES Y REDES DE INFORMACIÓN

TRANSPARENCIA Y PROTECCIÓN DE DATOS EN APLICACIONES MÓVILES EN ECUADOR: EVALUACIÓN DE LA DIFUSIÓN DE POLÍTICAS DE PROTECCIÓN DE DATOS EN APLICACIONES MÓVILES

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN TECNOLOGÍAS DE LA INFORMACIÓN

WASHINGTON ZAHID COPARA SUÁREZ

washington.copara@epn.edu.ec

DIRECTOR: DANNY SANTIAGO GUAMÁN LOACHAMÍN

danny.guaman@epn.edu.ec

Quito, agosto de 2024

AVAL

Certifico que el presente trabajo fue desarrollado por Washington Zahid Copara Suárez bajo mi supervisión.

_____

PhD. Danny S. Guamán

DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Washington Zahid Copara Suárez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

_____

WASHINGTON ZAHID COPARA SUÁREZ

DEDICATORIA

A mis padres y hermano. Porque ellos son los principales responsables de que pueda cosechar éxitos en mi vida. Su constante apoyo y confianza han sido fundamentales en cada paso que he dado. Estoy profundamente agradecido por su presencia y por lo que han hecho para que llegara hasta aquí.

AGRADECIMIENTO

A mis padres, Washington y Mónica, quienes con su sabiduría y consejos me han ayudado a mantenerme enfocado y alcanzar mis metas. Su apoyo incondicional y guía han sido fundamentales en cada paso de mi camino.

A mi hermano, Alexis, quien con su carisma y compañía me ha apoyado a lo largo de todo este tiempo. Su presencia constante y su ánimo han sido una fuente de fortaleza y motivación para mí.

A mis primos, porque, a pesar del tiempo que ha pasado, hemos continuado disfrutando juntos de grandes momentos en la vida. Su compañía y amistad han sido una parte valiosa de mi viaje.

A mis amigos de toda la vida, con quienes he podido avanzar incluso en los momentos más difíciles. Mención a Zahid Álvarez, uno de mis mejores amigos en estos tiempos, me brindó apoyo y gran amistad.

A mi tutor, Danny Guamán, por su orientación durante el desarrollo de este trabajo de titulación y por los valiosos consejos que me ha impartido.

ÍNDICE DE CONTENIDO

RESUMEN

La recolección de datos personales ha aumentado con la proliferación de dispositivos móviles, llegando a un 55.6% de la población ecuatoriana. Estos dispositivos recopilan una gran cantidad de datos con poca transparencia, lo que pone en riesgo la privacidad. La Ley Orgánica de Protección de Datos Personales (LOPDP), vigente desde mayo de 2021, regula el tratamiento de estos datos para equilibrar el uso legítimo de la información y la protección de la privacidad. La LOPDP exige a las organizaciones implementar políticas de protección de datos y sanciona su incumplimiento con multas que varían entre el 0.1% y el 0.7% del volumen de negocio.

Este trabajo de titulación tiene como propósito determinar si las organizaciones que proveen servicios a través de aplicaciones móviles a usuarios ecuatorianos cumplen con la difusión de las políticas de protección de datos, tal como lo estipula la LOPDP.

Para ello, en primer lugar, se revisa el marco teórico relacionado con la normativa de protección de datos personales en Ecuador, con énfasis en los requisitos de transparencia en la difusión de las políticas de protección de datos.

Posteriormente, se establecen criterios para evaluar estas políticas de privacidad. Una vez definidos los criterios, se identifican los datos necesarios para analizar su cumplimiento. Con los datos definidos, se implementan o adaptan artefactos de software para obtener información relevante para el análisis de cumplimiento. A continuación, se configura el entorno, se determina el número de aplicaciones y se clasifican en categorías potencialmente sensibles, incluyendo tanto aplicaciones internacionales como ecuatorianas.

Finalmente, se realiza el análisis del cumplimiento basado en toda la información recopilada.

PALABRAS CLAVE: datos personales, protección de datos, android, aplicaciones móviles, LOPDP, privacidad, cumplimiento.

ABSTRACT

The collection of personal data has increased significantly with the proliferation of mobile devices, reaching 55.6% of the Ecuadorian population. These devices collect a vast amount of data with little transparency, putting privacy at risk. The Organic Law for the Protection of Personal Data (LOPDP), in effect since May 2021, regulates the processing of this data to balance the legitimate use of information and the protection of privacy. The LOPDP requires organizations to implement data protection policies and penalizes non-compliance with fines ranging from 0.1% to 0.7% of their turnover.

This thesis aims to determine whether organizations that provide services through mobile applications to Ecuadorian users comply with the dissemination of data protection policies, as stipulated by the LOPDP.

To achieve this, the theoretical framework related to personal data protection regulations in Ecuador is first reviewed, with an emphasis on the transparency requirements for the dissemination of data protection policies.

Subsequently, criteria for evaluating these privacy policies are established. Once the criteria are defined, the necessary data for analyzing their compliance is identified. With the data defined, software artifacts are implemented or adapted to obtain relevant information for the compliance analysis. Next, the environment is configured, the number of applications is determined, and they are classified into potentially sensitive categories, including both international and Ecuadorian applications.

Finally, the compliance analysis is carried out based on all the information collected.

KEYWORDS: personal data, data protection, android, mobile apps, privacy, LOPDP, compliance.

1 INTRODUCCIÓN

En la actualidad, la recolección de datos personales se ha intensificado debido principalmente a la proliferación de los dispositivos móviles y sus aplicaciones [1]. En el Ecuador, aproximadamente el 55.6% de ciudadanos tiene un teléfono móvil [2]. Los teléfonos móviles y sus aplicaciones permiten el acceso a una amplia cantidad de datos personales, incluyendo la geolocalización, datos de contacto (p.ej., correos electrónicos, números de teléfono, direcciones), (pseudo) identificadores, entre otros. Lamentablemente, la recolección y su posterior tratamiento no siempre son transparentes para los usuarios [3], lo cual puede poner en riesgo su privacidad.

La Ley Orgánica de Protección de Datos Personales (LOPDP) [4], que entró en vigor el 26 de mayo de 2021, busca justamente normar el tratamiento de los datos personales por parte de las organizaciones. Esta persigue salvaguardar los datos personales de los ciudadanos, estableciendo pautas y requisitos claros para su manejo y protección por parte de las organizaciones, para, en última instancia, garantizar un equilibrio entre el uso legítimo de la información y la preservación de la privacidad [5].

Las políticas de protección de datos, también conocidas como políticas de privacidad, son el mecanismo de facto para brindar transparencia. A través de estas políticas, las organizaciones deberían informar a los usuarios sobre los diferentes tratamientos que efectuarán sobre los datos personales. Justamente una de las obligaciones del responsable del tratamiento de los datos personales, según el artículo 47 (literal 4) de la LOPD, es "*Implementar políticas de protección de datos personales afines al tratamiento de datos personales en cada caso en particular*". En caso de no hacerlo, el artículo 67 (literal 3) de la misma normativa, lo señala como una infracción leve, a la que podría aplicarse una sanción económica de entre el 0.1% y el 0.7% calculada sobre el volumen del negocio.

Las organizaciones públicas y privadas que manejan datos de carácter personal tuvieron un plazo de dos años para prepararse y prevenir incumplimientos. Desde el 26 de mayo de 2023, el régimen sancionatorio ya está en vigor por lo que las organizaciones que incumplan la LOPD pueden ser multadas. Pero, lamentablemente, un estudio del 31 de enero de 2023 estimó que menos del 1 % de las organizaciones estaban preparadas para prevenir potenciales incumplimientos con la ley [6].

En este contexto, el presente trabajo de titulación tiene como objetivo llevar a cabo un estudio cuantitativo sobre el grado de cumplimiento de las organizaciones respecto al requisito relacionado con la difusión de las políticas de protección de datos personales, establecidos en la Ley Orgánica de Protección de Datos Personales. El estudio se centra en aplicaciones móviles con sistema operativo Android, dado que este domina el mercado ecuatoriano con una participación del 88.3% [7].

1.1 Objetivo General

Evaluar el cumplimiento de la difusión de políticas de protección de datos por parte de las organizaciones que brindan servicios mediante aplicaciones móviles en Ecuador.

1.2 Objetivos Específicos

- Revisar el fundamento teórico pertinente, incluyendo los artículos de la Ley Orgánica de Protección de Datos Personales en Ecuador que sean relevantes para este trabajo y aportes de investigaciones previas.
- Establecer criterios de selección que aseguren una muestra representativa de aplicaciones, considerando factores como popularidad, categoría y ubicación geográfica de la organización.
- Identificar un conjunto de criterios de evaluación de cumplimiento de difusión de políticas de protección de datos según los artículos pertinentes de la LOPDP.
- Desarrollar o adaptar los artefactos de software que permitan obtener de manera automática los datos y metadatos desde la Play Store, necesarios para el análisis y evaluación del cumplimiento.
- Realizar un análisis con base en los datos recopilados para identificar tendencias y posibles correlaciones entre el cumplimiento y diferentes variables.

1.3 Alcance

Para evaluar el cumplimiento, se consideran las aplicaciones Android que ofrecen servicios a ciudadanos ecuatorianos. Inicialmente, se determinó una muestra de al menos 500 aplicaciones móviles Android. Sin embargo, debido a la cantidad de aplicaciones realmente representativas en Ecuador, se ajustó la cantidad de aplicaciones móviles a 379. Esta selección se realizó considerando criterios como la categoría de la aplicación, su popularidad y la ubicación geográfica de la organización.

Se implementan y adaptan los artefactos de software necesarios para descargar automáticamente los datos y metadatos desde la Play Store, incluyendo los permisos solicitados por la aplicación, número de descargas, desarrollador, categoría y URL de política de privacidad. Los permisos solicitados son usados como indicador del tipo de datos (personales) que podría solicitar una aplicación. Para descargar el texto de la política de privacidad se emplea el microservicio desarrollado en un trabajo previo "*Desarrollo de microservicios para la extracción de políticas de privacidad desde artefactos de una aplicación móvil Android*." [15].

Se elabora un informe de resultados del análisis cuantitativo de cumplimiento, centrados en el Artículo 47, literal 4, de la Ley Orgánica de Protección de Datos Personales. También se incluyen los hallazgos de potenciales patrones o correlaciones, por ejemplo, entre el (in)cumplimiento y la categoría de las aplicaciones

Se desarrollan un total de cinco módulos diseñados para proporcionar la información necesaria para verificar el potencial cumplimiento de los requisitos de transparencia en la divulgación de las políticas de privacidad según la Ley Orgánica de Protección de Datos, a continuación, una breve descripción sobre cada uno de los módulos:

- **Primer Módulo.** Permite la obtención automática de los documentos de las políticas de privacidad de las aplicaciones móviles a partir de las URLs correspondientes.
- **Segundo Módulo.** Permite la clasificación de las aplicaciones móviles en si recolectan o no datos personales.
- **Tercer Módulo.** Permite determinar si las aplicaciones móviles requieren de permisos sensibles por parte de los usuarios.
- **Cuarto Módulo.** Permite determinar si una política de privacidad cumple con los requisitos necesarios para ser considerada como tal. Para ello, se utiliza la Inteligencia Artificial Gemini de Google, que, mediante instrucciones específicas, proporciona una evaluación sobre si el documento cumple o no con las características de una política de privacidad.
- **Quinto Módulo.** Permite obtener los resultados finales sobre el potencial cumplimiento de la Ley Orgánica de Protección de Datos, basándose en la información recopilada en los módulos anteriores. La clasificación final determina si las aplicaciones cumplen o no con la normativa.

1.4 Marco Teórico

En esta sección se presentan los temas más relevantes relacionados con el trabajo de titulación. Estos temas abarcan aspectos de privacidad y datos personales, la Ley Orgánica de Protección de Datos Personales, requerimientos de transparencia, el análisis de políticas de protección de datos, y la privacidad y protección de datos en aplicaciones móviles en el contexto del sistema operativo Android. Además, se describen las distintas tecnologías y herramientas utilizadas implementar los módulos que permiten la obtención de datos para el análisis cuantitativo.

1.4.1 Privacidad y datos personales

En la actualidad, con el acelerado desarrollo de las tecnologías y los medios digitales, el concepto de privacidad ha adquirido una gran importancia. Este fenómeno se debe, principalmente, a que continuamente se utilizan recursos digitales que exigen de datos personales con el fin de 'mejorar' la experiencia de usuario. La recolección de datos es innegable, estas van desde las redes sociales, donde se comparte una gran variedad de datos personales, hasta el uso constante de los dispositivos móviles, que gracias a sus sensores pueden almacenar información sensible como los datos médicos [9].

Esta realidad ha dado lugar al término de 'derecho a la privacidad', el cual busca proteger la libertad de las personas de intrusiones o perturbaciones en su vida privada y asuntos personales [8].

Tomando en cuenta lo previamente descrito, según la Ley Orgánica de Protección de Datos [4], se puede definir a los datos personales como, "*dato que identifica o hace identificable a una persona natural, directa o indirectamente*", estos pueden ser: nombres, ubicación, información de contacto, número de teléfono, entre otros [9]. Además, según la LOPDP, se puede abarcar una variedad de categorías específicas, tales como:

- Datos biométricos: "*Dato personal único, relativo a las características físicas o fisiológicas, o conductas de una persona natural que permita o confirme la identificación única de dicha persona, como imágenes faciales o datos dactiloscópicos, entre otros*" [4].
- Datos genéticos: "*Dato personal único relacionado a características genéticas heredadas o adquiridas de una persona natural que proporcionan información única sobre la fisiología o salud de un individuo*" [4].

- Datos personales crediticios: "*Datos que integran el comportamiento económico de personas naturales, para analizar su capacidad financiera*" [4].
- Datos relativos a la salud: "*Datos personales relativos a la salud física o mental de una persona, incluida la prestación de servicios de atención sanitaria, que revelen información sobre su estado de salud.*" [4].

Esta información es la que se pretende proteger y mantener privada, y únicamente los usuarios deberán tener la potestad de decidir qué, con quién y cuando compartir sus datos personales.

Desde el punto de vista de la tecnología se abordan varios paradigmas de privacidad como: confidencialidad, control y transparencia [18].

1.4.1.1 Privacidad como confidencialidad

Este paradigma se centra en recopilar y exponer la menor cantidad de datos necesaria para que un usuario pueda acceder a un servicio de manera adecuada. En este tipo de privacidad se basa en el principio de que el usuario no confía plenamente en la organización que ofrece los servicios. Por lo tanto, el usuario intenta proteger su información al acceso no autorizado mediante técnicas de criptografía, cifrando todo tipo de información incluyendo los metadatos de sus archivos para que su acceso sea únicamente por el mismo usuario. La criptografía más común es la llamada punto a punto, ya que se busca que exista la confidencialidad de los datos entre ambos extremos [18].

1.4.1.2 Privacidad como control

Paradigma basado en el concepto de integridad contextual, que sugiere que durante el flujo de información se debe ajustar a normas específicas a un contexto dado, para mantener la privacidad. Es decir, el contexto es quien dicta las normas para determinar qué datos deben ser compartidos o expuestos. Por ejemplo: Si se tiene una aplicación móvil de salud puede ser necesario compartir los datos del paciente hacia otras organizaciones del mismo campo, por lo que en este contexto se podría transferir la información, aunque no a terceros no autorizados [18].

La información puede ser entregada de manera contractual o por consentimiento. En el primer caso es necesario que el usuario entregue ciertos datos personales para poder utilizar el servicio caso contrario no podrá hacerlo. Con respecto al consentimiento, este se basa principalmente en que el usuario puede entregar de manera consciente y voluntaria

sus datos o negarse a hacerlo sin que afecte el funcionamiento del servicio al que se está suscribiendo [18].

1.4.1.3 Privacidad como transparencia

La transparencia se refiere a la claridad y apertura en el manejo de datos personales, permitiendo a los usuarios comprender y controlar cómo se utiliza su información. Existen dos mecanismos principales de transparencia: el primero se basa en la retroalimentación, proporcionando a los usuarios información sobre el uso de sus datos, y el segundo en auditorías, que permiten evaluar y verificar el cumplimiento de las prácticas de manejo de datos [18].

La retroalimentación dictamina que el usuario puede ver como su información es procesada y percibida por los demás. El ejemplo más común es el de Facebook, ya que este permite controlar la privacidad y que se quiere mostrar al público, en este caso puede mostrar un perfil más completo hacia los conocidos o amigos y oculto para los desconocidos, este mecanismo se conoce como espejo [18].

El mecanismo basado en auditorías funciona bajo el hecho de que se requiere que los sistemas involucrados en los servicios registren todas las operaciones de acceso y procesamiento de datos. Este registro puede revelar cierta información como: los inicios de sesión de los usuarios, transmisión de datos, quién y con que propósito han accedido a los datos del usuario, entre otros. Estos registros permitirán obtener finalmente un informe de las brechas de seguridad del sistema para así poder tomar acciones pertinentes para asegurar la privacidad [18].

1.4.2 Ley orgánica de protección de datos personales

La LOPDP fue publicada en el Registro Oficial Suplementario No. 459 el 26 de mayo de 2021. "*El principal objetivo y finalidad de esta ley es el poder garantizar el ejercicio del derecho a la protección de datos personales, incluyendo el acceso y decisión sobre la información y datos de este carácter, así como su correspondiente protección*" [4].

El Capítulo 1, en sus artículos 7 y 8, aborda dos aspectos fundamentales: el tratamiento legítimo de los datos personales y el consentimiento. Estos artículos subrayan la importancia de que el titular de los datos debe otorgar su consentimiento para que sus datos personales sean tratados, especificando una o varias finalidades concretas.

El capítulo 2 menciona los principios por los que se regirá la ley. Entre ellos se encuentran:

- Transparencia
- Finalidad
- Confidencialidad
- Consentimiento
- Seguridad de datos personales

En el presente trabajo, la transparencia es uno de los principios más importantes a considerar, dado que el objetivo principal es evaluar que tan transparentes son las empresas ecuatorianas y extranjeras con base en Ecuador. Este determina si se cumple con el principio de que el tratamiento de datos (política de privacidad) debe ser fácilmente accesible y comprensible, utilizando un lenguaje claro y sencillo.

Además, el capítulo 3 de la LOPDP aborda los derechos, y en su artículo 12 menciona el derecho a la información. Este derecho garantiza que el usuario siempre esté informado sobre diversos aspectos, destacando entre los más importantes.:

- Fines del tratamiento de datos
- Base legal del tratamiento
- Tipos de tratamiento
- Existencia de una base de datos donde se almacenen sus datos personales

Este artículo, al igual que los del capítulo 1 de la LOPDP, permitirá definir ciertos criterios de evaluación del cumplimiento de las aplicaciones móviles.

1.4.3 Requerimientos de transparencia

La LOPDP del Ecuador establece a la transparencia como "*toda información o comunicación relativa al tratamiento de datos personales deberá ser fácilmente accesible y fácil de entender y se deberá utilizar un lenguaje sencillo y claro*" [4].

Adicional en el artículo 12 del capítulo 3, se mencionan ciertos criterios importantes con respecto al tratamiento de datos personales y la información que el usuario tiene derecho a saber, incluyendo los siguientes [4]:

- Identidad y datos de contacto del responsable del tratamiento de datos personales, que incluirá: dirección del domicilio legal, número de teléfono y correo electrónico.
- Cuando sea del caso, identidad y datos de contacto del delegado de protección de datos personales, que incluirá: dirección domiciliaria, número de teléfono y correo electrónico.

- Las consecuencias para el titular de los datos personales de su entrega o negativa a ello.
- La posibilidad de revocar el consentimiento.
- Tiempo de conservación.
- La existencia y forma en que pueden hacerse efectivos sus derechos de acceso, eliminación, rectificación y actualización, oposición, anulación, limitación del tratamiento y a no ser objeto de una decisión basada únicamente en valoraciones automatizadas.

Esta información, junto con las prácticas de procesamiento de datos (recolección, almacenamiento y uso), se definen en políticas de protección de datos, que deben informarse oportunamente a los ciudadanos. En este sentido, el artículo 47 de la LOPDP establece que el responsable y encargado del tratamiento de datos personales debe "*Implementar políticas de protección de datos personales afines al tratamiento de datos personales en cada caso en particular*" [4]. Estas políticas deben diseñarse para satisfacer el principio definido de transparencia.

1.4.4 Análisis de las políticas de protección de datos

Las políticas de protección de datos, también conocidas como políticas de privacidad, son documentos que las organizaciones proporcionan a los usuarios de sus aplicaciones móviles. El objetivo de estas políticas es informar a los usuarios sobre los datos personales que la organización recopilará, además, de cómo y para qué se utilizarán [4].

Aunque el contenido de las políticas de privacidad puede variar según su base legal, existen características comunes identificadas en [10], entre los que se destacan:

- La importancia de que las políticas de privacidad sean fáciles de leer y entender, debido a que muchos usuarios no comprenden los riesgos de privacidad a los que se enfrentan.

- Los datos recopilados deben tener coherencia con respecto al propósito o el fin del producto o servicio que ofrece la organización. Esto es conocido como la evaluación de coherencia.

- Informar al usuario sobre algunas prácticas declaradas en el texto de privacidad, entre las que se puede encontrar, tipos de datos personales, finalidad de la recopilación de datos, compartición de datos, derechos del usuario de acceder, rectificar o eliminar datos y medidas de seguridad para protegerlos, entre otros.

Adicionalmente, en el ANEXO A se presenta una tabla que contiene la información esencial que debe incluir una política de privacidad. Esta información abarca el identificador de las organizaciones, los datos personales recopilados, las operaciones realizadas sobre estos datos, el propósito explícito del procesamiento, el consentimiento de los usuarios, el acceso de los usuarios a su información, el tiempo de retención de los datos, las medidas de seguridad implementadas, los cambios en la política de privacidad y aspectos relacionados con la protección de datos de menores. [10].

La importancia de analizar las políticas de protección de datos radica en evaluar los requisitos de transparencia establecidos por la LOPDP. Este análisis puede llevarse a cabo de forma manual o mediante herramientas de inteligencia artificial.

La principal ventaja que se obtiene con la inteligencia artificial es que se pueden analizar muchas políticas de privacidad rápidamente aminorando significativamente los tiempos con respecto a un análisis manual. Recientemente, la Inteligencia Artificial Generativa (GenAI), principalmente aquellas basadas en los Modelos de Lenguaje Grande (LLM), es capaz de comprender y generar texto similar a como lo haría un humano, lo cual lo hace ideal para poder realizar la compleja tarea de analizar las políticas de privacidad [11].

En el estudio realizado en [11], se establece que para obtener los mejores resultados es necesario escribir o diseñar una instrucción adecuada. La estructura de una instrucción básica está dividida en tres partes: datos, tarea y formato de salida, ver Figura 1.1. Los datos lo conforman el texto de una política de privacidad que se va a analizar. La tarea vendría a ser la práctica específica (recolección, transferencia, retención, etc.) que se quiere identificar en la política de privacidad. Finalmente, el formato de salida define cómo se desea que sea la respuesta por parte de la IA.

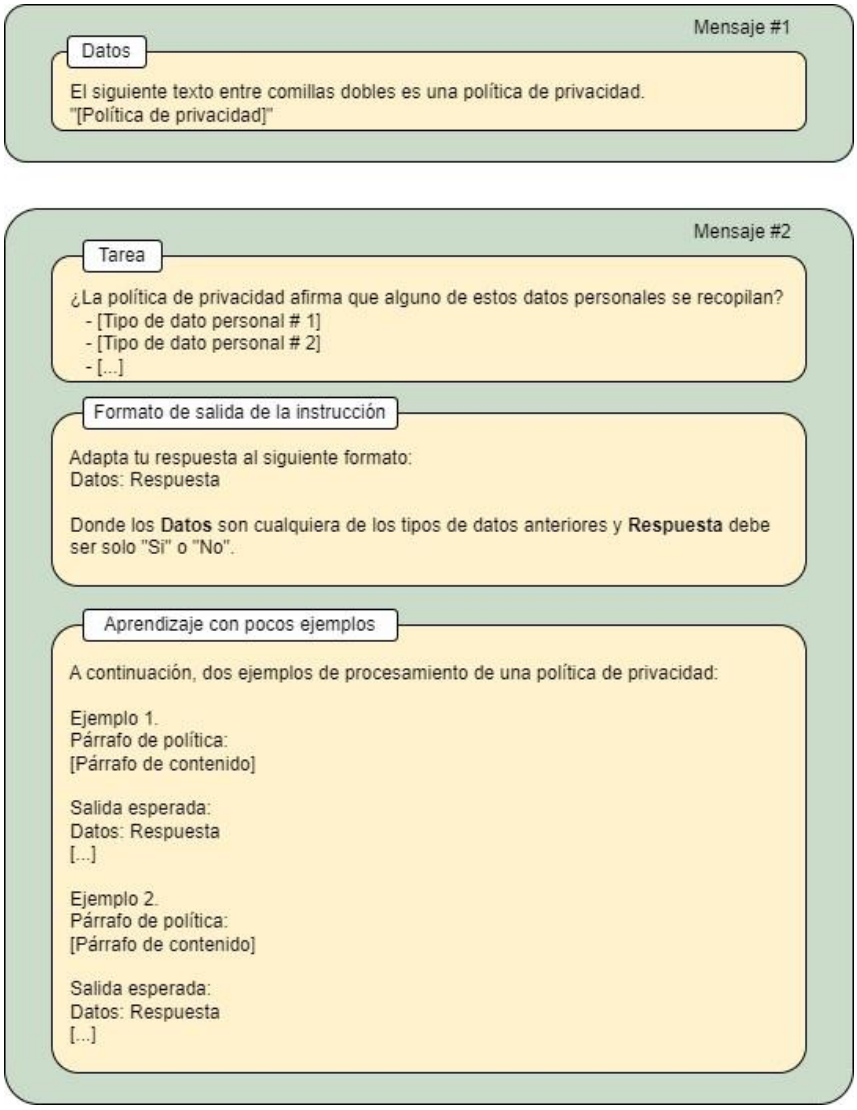

Figura 1.1 Diseño de la instrucción final [11].

1.4.5 Aplicaciones móviles (Android OS)

Android es un sistema operativo basado en Linux, inicialmente diseñado para teléfonos móviles. Actualmente, este sistema funciona en unos 2500 millones de dispositivos, incluyendo teléfonos móviles, tabletas, relojes inteligentes, automóviles y televisores [12].

Con respecto al sistema operativo móvil, según datos de junio del 2024, Android lidera la cuota de mercado de teléfonos móviles con un 72,15%, mientras que iOS de Apple representa el 27,19% [13]. La tienda de aplicaciones de Android se denomina Google Play, el cual es un espacio digital para el almacenamiento de las aplicaciones móviles del sistema operativo. En este espacio, los usuarios pueden realizar búsquedas mediante el nombre de la aplicación, categorías u otros filtros como número de descargas, recientemente añadidas, entre otros. Hasta julio de 2024 se encuentran 1.700.526 aplicaciones móviles

[14]. En Ecuador, el sistema operativo Android domina el mercado de dispositivos móviles, con aproximadamente el 88% de la cuota de mercado [15].

1.4.5.1 Permisos en Android

Las aplicaciones móviles en Android cuentan con niveles de permisos específicos para regular el acceso tanto al hardware (sensores) como al software (otras aplicaciones del smartphone). Estos permisos se definen en el archivo AndroidManifest.xml de la aplicación. Según la documentación oficial para el desarrollo de aplicaciones en Android, existen 206 permisos activos para acceder a recursos y datos del dispositivo móvil [16].

1.4.5.2 Tipos de permisos básicos

En el modelo de permisos gestionado por Android se definen 4 tipos de permisos [15]:

- Normal: Permisos de bajo riesgo que se otorgan durante la instalación de la aplicación sin la aprobación explícita del usuario.
- Peligroso (Dangerous): Permisos de alto riesgo que son solicitados explícitamente por la aplicación para acceder a los datos del usuario o para controlar el dispositivo.
- Firma (Signature): Son permisos que se otorgan automáticamente únicamente si la aplicación solicitante está firmada con el mismo certificado del paquete de software que declara el permiso.
- Firma o sistema (Signature or system): Permisos otorgados solo a las aplicaciones que tienen una carpeta dedicada en la imagen del SO de Android o que están firmadas con el mismo certificado que la aplicación que declara el permiso.

Para los fines de este trabajo, solo se definirán los permisos considerados sensibles, que corresponden principalmente a los permisos básicos de Android clasificados como 'Peligrosos'. En el ANEXO B se incluye una lista de estos permisos, entre los que se abarca aquellos relacionados con la recepción de mensajes y llamadas, la multimedia, el acceso a sensores, la ubicación, ya sea exacta o aproximada, entre otros [17].

1.4.5.3 Categorías de aplicaciones en Google Play

En Google Play existen varias categorías de aplicaciones. Sin embargo, para los propósitos del presente trabajo de titulación, se identifican únicamente aquellas que tienen una mayor probabilidad de procesar datos personales de los usuarios. A continuación, se presentan las categorías seleccionadas y la razón detrás de su elección:

- Salud: Esta categoría reviste una importancia crucial debido a que implica el manejo de datos médicos, los cuales pueden ser considerados altamente sensibles por su naturaleza. Por lo tanto, es de suma importancia analizar qué tipos de permisos se solicitan y qué datos se procesan en las distintas aplicaciones de esta categoría.
- Financiero: Las aplicaciones móviles de finanzas, principalmente desarrolladas por bancos, procesan una gran cantidad de información personal, necesaria para la apertura de cuentas y otros procesos. Es importante considerar qué permisos solicitan y qué datos personales se están procesando.
- Social: En esta categoría, se puede acceder a información privada y personal no solo de los usuarios directos, sino también de los contactos de los usuarios. Entre las principales plataformas se encuentran Facebook, Instagram y otras similares.
- Entretenimiento: Aquí se incluyen aplicaciones relacionadas a streaming de video, información de películas, juegos, entre otras. En este contexto, es posible encontrar información personal de los usuarios, como sus preferencias de entretenimiento, tales como series o películas.
- Videojuegos: Esta categoría presenta un alto riesgo, especialmente porque está dirigida principalmente a un público infantil y juvenil. Por lo tanto, el almacenamiento y procesamiento de información en este ámbito es extremadamente sensible en términos de privacidad.

1.4.5.4 Seguridad de datos en Google Play

La seguridad de los datos en Google Play es una sección importante en donde el desarrollador de las aplicaciones pone en evidencia cuales son aquellos tipos de datos que se podrían recopilar o compartir. En Google Play se presentan principalmente dos secciones importantes [38]:

i. **Etiquetas de privacidad:** Las cuales señalan los datos que podría estar recopilando la aplicación móvil. En la Figura 1.2, se presenta un ejemplo de cómo se observa la etiqueta de privacidad en Google Play.

Figura 1.2 Ejemplo de etiqueta de privacidad de Google Play [38].

ii. **URL de la política de privacidad:** En la sección de seguridad de datos de Google Play, se incluye una subsección que presenta la política de privacidad de cada aplicación móvil, proporcionada por el desarrollador. Esta presentación de la política de privacidad se puede apreciar en la Figura 1.3.

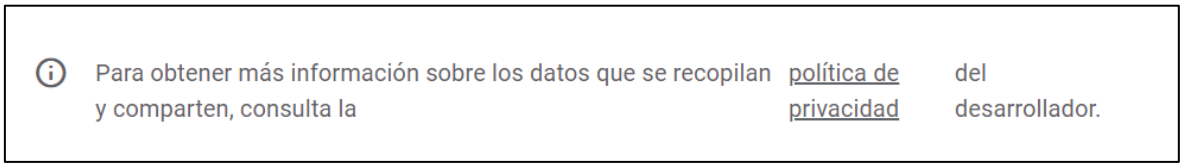

Figura 1.3 Ejemplo de URL de política de privacidad presentada en Google Play [38].

Adicionalmente, dentro de la propia aplicación móvil, se definen los permisos que esta requiere de los usuarios. Estos permisos forman parte de los metadatos de la aplicación y son fundamentales, ya que están directamente relacionados con la seguridad de los datos. Esto debido a que, es esencial que los usuarios conozcan los permisos que las aplicaciones móviles les solicitan. En la Figura 1.4, se muestra un ejemplo de cómo se observan los metadatos de los permisos dentro de una aplicación móvil Android.

```
1  <uses-permission android:name ="android.permission.INTERNET" />
2  <uses-permission android:name ="android.permission.ACCESS_NETWORK_STATE" />
3  <uses-permission android:name ="android.permission.VIBRATE" />
```

Figura 1.4 Ejemplo de los metadatos en una aplicación Android.

1.4.6 Privacidad y protección de datos en aplicaciones móviles

Los riesgos a la privacidad de un usuario se exacerban en el ecosistema de las aplicaciones móviles, que se han convertido en un dispositivo imprescindible para un usuario y que lo lleva a todas partes. Los usuarios confían en gran medida en las aplicaciones móviles al proporcionar su información personal, a menudo sin comprender completamente cómo se manejan sus datos.

El funcionamiento de estas aplicaciones puede ser complejo debido a su entorno dinámico, ya que a menudo reutilizan bibliotecas de software y se conectan a diversas redes y sistemas. Como resultado, evaluar sus características de privacidad y seguridad se vuelve más difícil.

Aunque algunas prácticas deficientes en protección de datos pueden parecer malintencionadas, en muchos casos se deben a una falta de conciencia, conocimiento o comprensión por parte de los desarrolladores sobre cómo integrar adecuadamente los requisitos de privacidad y seguridad en sus herramientas [32]. Esto conlleva riesgos para la privacidad y la protección de datos en las aplicaciones móviles, entre los que destacan:

i. Variedad de datos y múltiples sensores

Los dispositivos móviles pueden acceder a varios datos personales y sensibles, como información sobre bienestar, salud o datos médicos, que los usuarios proporcionan mediante las aplicaciones. Además, estos dispositivos incorporan una gran variedad de sensores como micrófono, cámara, acelerómetro, GPS, Wifi, entre otros.

Estos sensores pueden generar una gran cantidad de datos y metadatos que se consideran personales tales como, ubicación, hora y temperatura. Con esto en consideración se pueden tener impactos inesperados con respecto a la privacidad. Por ejemplo, se ha demostrado como el uso de sensores de movimiento, acelerómetro o giroscopio permiten el identificar y autenticar fácilmente a un usuario [32].

ii. Dispositivo personal, siempre "encendido"

Los usuarios de los dispositivos móviles tratan a sus equipos como una extensión de sí mismos, ya que lo consideran como algo de mucha confianza que nunca compartirán con nadie. Estos dispositivos, siempre encendidos y llevados a todas partes, permiten una gran recopilación de datos durante mucho tiempo. Esto conlleva a que sean objetos perfectos para una vigilancia omnipresente y continua de los usuarios. Esta vigilancia se la denomina como 'líquida' debido a que hasta los más pequeños detalles de la vida de los usuarios puede ser rastreada y registrada [32].

iii. Diferentes tipos de identificadores

Los dispositivos móviles pueden contener una amplia variedad de identificadores, como el ID del hardware del dispositivo, archivos almacenados, metadatos y huellas digitales. Estos identificadores pueden ser utilizados por las aplicaciones móviles para identificar y rastrear a los usuarios. Según [32], utilizando solo cuatro puntos espaciotemporales obtenidos de un dispositivo móvil, es posible identificar de manera única al 95% de las personas [35]. Además, se ha comprobado que cualquier combinación de cuatro aplicaciones móviles es suficiente para identificar a un usuario en el 95% de los casos [36]. Es importante tener en cuenta que muchos identificadores, como la huella digital, son permanentes y extremadamente difíciles de modificar.

1.4.7 Tecnologías y herramientas empleadas en este proyecto

En esta sección se van a describir las tecnologías y herramientas que han sido utilizadas durante este trabajo de titulación.

1.4.7.1 Python

Python es un lenguaje de programación ampliamente utilizado debido a su eficiencia y la facilidad de su aprendizaje, además, puede ser utilizado en distintas plataformas. Este lenguaje permite una variedad de aplicaciones, entre ellas, aplicaciones web, desarrollo de software, ciencia de datos y aprendizaje automático. Este lenguaje de programación es libre y se integra bien a todos los tipos de sistemas [20].

Los principales beneficios que ofrece Python son los siguientes:

- Debido a su sintaxis básica similar a la del inglés, los desarrolladores tienen la capacidad de leer y comprender fácilmente los programas de Python.
- Python permite a los desarrolladores escribir código con menos líneas en comparación a otros lenguajes de programación, lo que mejora la productividad de los programadores.
- Python cuenta con una gran variedad de bibliotecas.
- Los desarrolladores que tienen experiencia con otros lenguajes de programación podrán utilizar Python de manera más sencilla.

1.4.7.2 Bibliotecas utilizadas

1.4.7.2.1 Pandas

Es una biblioteca de código abierto diseñado específicamente para la manipulación de datos en el lenguaje de Python. Esta biblioteca permite cargar, alinear, manipular e incluso fusionar datos. El rendimiento de Pandas es impresionante debido a que puede manipular una gran cantidad de datos [21].

1.4.7.2.2 Openpyxl

Esta biblioteca permite leer y escribir archivos Excel a partir de la versión 2010, incluyendo las extensiones de archivo .xlsx, .xlsm, .xltx y .xltm [22]. Se integra perfectamente con Pandas, facilitando la exportación de resultados directamente en archivos Excel. Esto ayuda a evitar la pérdida de datos o confusiones que podrían surgir, por ejemplo, al convertir archivos CSV.

1.4.7.2.3 Google Generative AI

Es una biblioteca desarrollada por Google, que permitirá la integración de la API de Gemini en los programas de Python. Esta biblioteca funciona principalmente con una llave de Google, esta es la que permitirá la conexión con la API de Google [23].

1.4.7.2.4 JSON

Esta biblioteca es utilizada principalmente para poder trabajar con datos JSON, permitiendo convertir entre datos de JSON a tipos de datos de Python [24].

1.4.7.3 Gemini

Gemini es un modelo de lenguaje grande (LLM) multimodal desarrollado por Google y su subdivisión Google AI. Este fue anunciado el 6 de diciembre del 2023, siendo en un principio el competidor directo de GPT-4 de OpenAI [25].

Gemini es un modelo multimodal con la capacidad de comprender diversos tipos de información, lo que lo convierte en un modelo extremadamente flexible. Puede entender texto, imágenes, audio e incluso lenguajes de programación. Su funcionamiento se basa principalmente en el entrenamiento intensivo con una gran cantidad de datos. Gemini recopila datos de internet y alimenta sus algoritmos de entrenamiento para aprender y generar respuestas escritas de manera natural [26].

Gemini tiene 3 versiones disponibles para su uso:

- Gemini 1.5 Flash
- Gemini 1.5 Pro
- Gemini 1.0 Pro

Cada una de las versiones tienen su plan gratuito, estos vienen con limitaciones. Sin embargo, pueden servir correctamente en el contexto del proyecto actual. En la Tabla 1.1, se detalla el plan gratuito de Gemini 1.0 Pro [27]:

Tabla 1.1 Plan Gratuito Gemini 1.0 Pro

Límites de frecuencia	15 RPM (Solicitudes por minuto)
	32.000 TPM (Tokens por minuto)
	1.500 RPD (Solicitudes por día)
Precio (entrada)	Sin costo
Almacenamiento de contexto en caché	No aplicable
Precio (resultados)	Sin costo

La diferencia principal con respecto a la API GPT de OpenAI es que Gemini permite tener un plan gratuito para poder realizar así pequeños proyectos, algo que la inteligencia artificial de OpenAI no permite. En la Tabla 1.2, se puede observar los precios del modelo GPT-3.5 que vendría a ser la equivalencia a Gemini 1.0 Pro [28].

Tabla 1.2 Planes de GPT-3.5 Turbo

Modelo	Precios
gpt-3.5-turbo-0125	0,50 US$ / 1 M Tokens de entrada
	1,50 US$ / 1 M Tokens de salida
gpt-3.5-turbo-instruct	1,50 US$ / 1 M Tokens de entrada
	2,00 US$ / 1 M Tokens de salida

Si bien GPT-3.5 ofrece un bajo costo de uso, Gemini se presenta como una alternativa atractiva debido a su simplicidad de implementación. Para comenzar a utilizar Gemini, solo se necesita generar una clave de proyecto en Google AI Studio. Además, es posible generar múltiples claves en caso de alcanzar el límite de uso del modelo, garantizando un trabajo continuo sin interrupciones [27].

A continuación, se va a presentar un ejemplo de cómo se puede implementar la API de Gemini en el código de Python [23].

i. Instalar la librería

En la Figura 1.5, se observa el comando para instalar la API de Google Gemini.

```
pip install -U google-generativeai
```

Figura 1.5 Instalación de librería de la API.

ii. Importar el SDK y configurar la clave de la API

En la Figura 1.6, se observa cómo se ha importado la API de Gemini, además de cómo se realiza la configuración de la clave de proyecto. Esta clave se debe obtener desde Google AI Studio y se pueden generar varias.

```
import google.generativeai as genai
import os

genai.configure(api_key=os.environ["GEMINI_API_KEY"])
```

Figura 1.6 Importación de SDK y clave de la API

iii. Crear el modelo y ejecutar la instrucción

En la Figura 1.7, se observa cómo se ha instanciado el modelo de inteligencia artificial de Gemini 1.0 Pro. Además, el ingreso de la instrucción en este caso es “Lo opuesto a caliente es”, para luego imprimir la respuesta.

```
model = genai.GenerativeModel('gemini-1.0-pro-latest')
response = model.generate_content("The opposite of hot is")
print(response.text)
```

Figura 1.7 Instancia de modelo y ejecución.

1.4.7.4 Microservicios

Los microservicios son un estilo de arquitectura y un modo de desarrollar software, permitiendo a las aplicaciones dividirse en elementos más pequeños e independientes entre sí. Cada uno de estos elementos o procesos es un microservicio. El principal objetivo de desarrollar bajo microservicios es el poder distribuir software de calidad con mayor rapidez. Los microservicios funcionan en conjunto con contenedores, debido a que permiten aprovechar mejor el sistema de hardware y facilitan la coordinación de los servicios, entre lo que se incluyen almacenamiento, conexión de red y seguridad [29].

1.4.7.5 Docker

Docker es un sistema de software de TI que posibilita la creación y el uso de contenedores en Linux. Tiene una comunidad de código abierto que se encarga de mejorar las tecnologías para el beneficio de todos los usuarios. Además, está respaldada bajo la empresa Docker Inc., la cual se basa en el trabajo de la comunidad para brindar soporte a las tecnologías mejoradas y reforzadas para clientes empresariales [30].

En la Figura 1.8, se observa como la tecnología Docker favorece a la división de las aplicaciones mediante sus procesos individuales además de las herramientas para hacerlo.

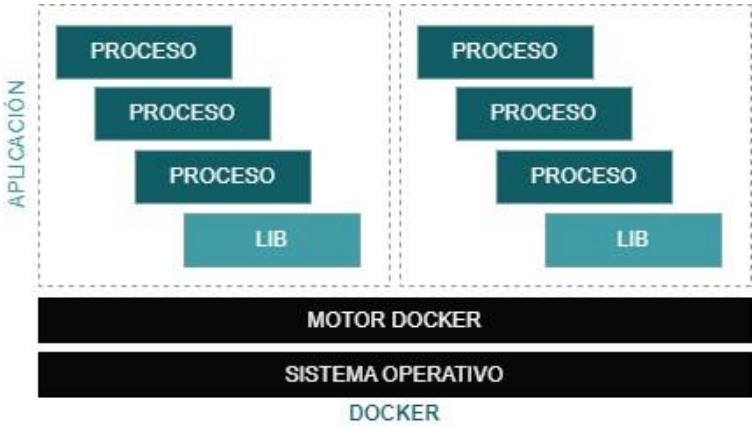

Figura 1.8 Estructura de Docker [30].

Docker permite utilizar contenedores como máquinas virtuales muy livianas y modulares. Además, facilita su creación, implementación, el poder copiarlos y trasladarlos entre varios entornos [30].

2 METODOLOGÍA

En este capítulo se presenta la metodología seguida en el desarrollo del presente trabajo de titulación. La Figura 2.1 ilustra esta metodología.

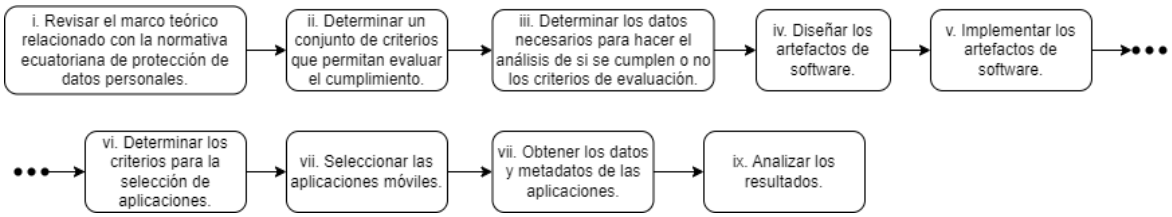

Figura 2.1 Fases de la metodología empleada en este trabajo de titulación

A continuación, se presenta una breve descripción de cada bloque:

i. **Revisar el marco teórico relacionado con la normativa de protección de datos personales. -** Esta fase aborda la revisión teórica de la Ley Orgánica de Protección de Datos Personales en Ecuador, incluyendo los requisitos de transparencia y el análisis de las políticas de privacidad. También se examinan temas relacionados con la privacidad y los datos personales desde las perspectivas legal, tecnológica y, específicamente, en el contexto de las aplicaciones móviles.

ii. **Determinar un conjunto de criterios que permiten evaluar el cumplimiento. -** en la siguiente fase se definen los criterios para evaluar el cumplimiento de las políticas de protección de datos por parte de las organizaciones que ofrecen servicios mediante aplicaciones móviles en Ecuador.

iii. **Determinar los datos necesarios para hacer el análisis de si se cumplen o no los criterios de evaluación. -** Esta fase está directamente relacionado con el anterior, ya que los criterios de evaluación proporcionan los datos necesarios para determinar el potencial cumplimiento. Además, posteriormente, estos datos se descargarán y clasificarán mediante artefactos de software.

iv. **Diseñar los artefactos de software. -** En esta fase se diseñan los artefactos de software que permiten obtener los datos y metadatos de las aplicaciones móviles para su posterior análisis.

v. **Implementar los artefactos de software. -** En esta fase se implementan o adecuan los artefactos de software previamente diseñados.

vi. **Determinar los criterios para la selección de aplicaciones. -** En esta fase se determinan ciertos criterios para la selección de las aplicaciones móviles, que más se puedan ajustar al propósito del presente trabajo.

vii. **Seleccionar las aplicaciones móviles.** Con base en los criterios anteriores, se busca y selecciona las aplicaciones que cumplan con los criterios definidos.

viii. **Obtener los datos y metadatos de las aplicaciones. -** En esta fase, se descargan los datos y metadatos de las aplicaciones móviles seleccionadas. Estos datos se organizan en hojas de cálculo, pues permite una mejor visualización de los resultados y su análisis posterior.

ix. **Analizar los resultados. -** En esta fase, con los datos y metadatos obtenidos, se realiza un análisis para verificar la tasa de cumplimiento en la difusión de políticas de datos por parte de las organizaciones que ofrecen servicios mediante aplicaciones móviles en Ecuador. Además, se examinan posibles tendencias y correlaciones con otras variables.

2.1 Análisis de requerimientos

2.1.1 Criterios para la evaluación del cumplimiento

Es fundamental cumplir con lo establecido por la Ley Orgánica de Protección de Datos Personales de Ecuador, en lo que respecta a la difusión de políticas de protección de datos por parte de las organizaciones que brindan servicios mediante las aplicaciones móviles del sistema operativo Android.

A continuación, se definen los dos criterios de evaluación identificados con base en la Ley Orgánica de Protección de Datos.

2.1.1.1 Criterio 1: La aplicación está procesando información de carácter personal

Todas las organizaciones que procesen datos personales deben cumplir con la Ley Orgánica de Protección de Datos Personales. Por lo tanto, el primer criterio consiste justamente en saber si una organización está recolectando datos personales a través de sus aplicaciones.

Según la ley orgánica de protección de datos, un dato personal se puede definir como "*Dato que identifica o hace identificable a una persona natural, directa o indirectamente*" [4].

Las aplicaciones móviles, dependiente de su tipo, pueden acceder a varios tipos de datos personales y sensibles, incluyendo [32]:

- Datos de bienestar
- Datos de salud y médicos

Por otro lado, las aplicaciones móviles pueden permitir a las organizaciones recolectar datos a través de sus formularios, incluyendo las siguientes:

- Nombre y apellidos
- Número de identificación (como número de identificación nacional o pasaporte)
- Identificador en línea (como dirección de correo electrónico o nombre de usuario)
- Características especiales que expresan la identidad física, fisiológica, genética, mental, comercial, cultural o social de una persona

Asimismo, debido a que los dispositivos móviles tienen una gran cantidad de sensores (p.ej., el GPS) y estructuras de datos internas (p.ej., la lista de contactos), podrían recolectar los siguientes datos de carácter personal:

- Datos de voz a través de los micrófonos
- Datos fotográficos a través de la cámara
- Datos de geolocalización a través del GPS
- Números de teléfono
- Cuentas de correo electrónico almacenadas
- Datos de tarjetas de crédito
- Números de cuenta bancaria
- Dirección postal
- Identificadores únicos (p.ej., IMEI, MAC)
- Pseudo identificadores (p.ej., IP, huella dactilar)

Si una aplicación procesa al menos uno de los tipos de datos presentados previamente, se puede decir que se ha cumplido con el criterio y que, por lo tanto, tiene que cumplir con los requisitos establecidos en la normativa, incluyendo la difusión de políticas de privacidad o protección de datos.

2.1.1.2 Criterio 2: El dueño de la aplicación móvil está difundiendo la política de privacidad

Según establece el artículo 12 de la Ley Orgánica de Datos Personales, todo usuario tiene derecho a ser informado sobre el uso de sus datos personales, generalmente a través avisos o políticas de privacidad. Dicha política debe ser compartida con el usuario a través

de cualquier medio, ya sea proporcionando la URL de la política en la tienda de aplicaciones directamente o dentro de la propia aplicación móvil.

Además, la política de privacidad debe cumplir con los principios de transparencia según lo establecido en el artículo 10 de la ley, el cual dicta que "*Toda información o comunicación relativa a este tratamiento deberá ser fácilmente accesible y comprensible, utilizando un lenguaje sencillo y claro*" [4].

En este contexto, se puede afirmar que una organización que está procesando datos personales y que comparte su política de privacidad estaría cumpliendo –parcialmente– con este requisito. Se dice parcialmente porque habría que verificar todavía si lo que dice la política de privacidad está en línea con lo que realmente hacen sus sistemas de información, incluyendo sus aplicaciones móviles. En este trabajo nos centraremos únicamente en evaluar si está difundiendo una política de privacidad, sin evaluar sus detalles internos o su calidad; esto último será parte de trabajos futuros.
Es importante destacar que, en ocasiones, las organizaciones publican URLs que son enlaces hacia los términos y condiciones de uso o hacia políticas empresariales, que no son realmente políticas de privacidad. En este sentido, en este trabajo evaluaremos sin en efecto el texto publicado se trata de una política de privacidad.

2.1.2 Determinar los datos necesarios para el análisis de los criterios de evaluación

Una vez establecidos los criterios de evaluación, se puede determinar qué datos son necesarios para evaluar su cumplimiento Basada en los datos y metadatos disponibles en Play Store, se han identificado los siguientes datos:

i. **Datos personales recopilados:** Se refiere a los datos que la aplicación móvil obtiene en general. Estos datos están disponibles en la etiqueta de privacidad publicada en Play Store y permitirán evaluar el criterio 1 antes descrito.
ii. **Datos personales compartidos:** Son los datos que las aplicaciones móviles recolectan y posteriormente comparten con terceros. Estos datos están disponibles en la etiqueta de privacidad publicada en Play Store y permitirán evaluar el criterio 1 antes descrito.
iii. **Permisos de la aplicación:** Esto se refiere a los permisos que requiere la aplicación para acceder a sensores (p.ej. GPS) y estructuras de datos (p.ej., lista de contactos) sensibles desde el punto de vista de la privacidad de los usuarios. En el Anexo A, se muestra una lista de estos permisos, que permitirán evaluar el criterio 1 antes descrito.

iv. **URL de la política de privacidad:** Se refiere a la URL de la política de privacidad que una organización está obligada a publicar en Play Store si procesa datos personales [37]. A través de esta URL se podrá descargar el texto publicado y verificar si en efecto se trata de una política de privacidad, para así evaluar el criterio 2 antes descrito.

2.2 Diseño del sistema

En esta sección se presentan los aspectos más relevantes del diseño de los artefactos de software, organizados en cinco módulos, con uno adicional proveniente de la colaboración con la Universidad Politécnica de Madrid. La Figura 2.2 muestra el diagrama de bloques que ilustra todos los módulos implementados.

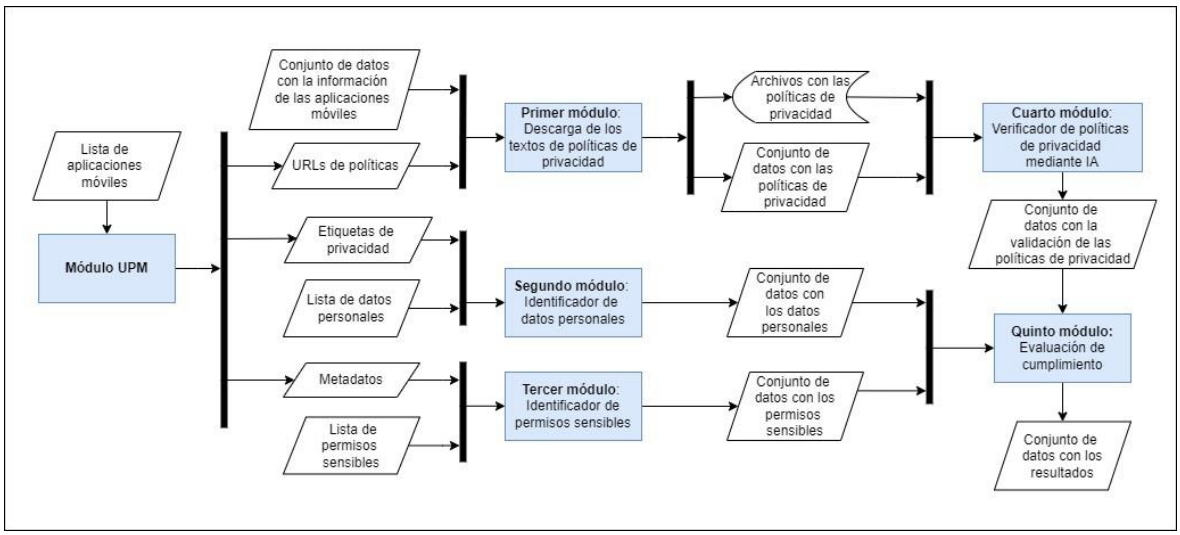

Figura 2.2 Diagrama de bloques que involucra los módulos del trabajo de titulación.

A continuación, se presenta una explicación de cada módulo explicando su principal funcionamiento.

i. **Módulo UPM:** Se utiliza la infraestructura de la Universidad Politécnica de Madrid para este proceso. Es necesario enviar una lista de las aplicaciones móviles de las que se requiere extraer información desde la Play Store. También se deben especificar los datos que se necesitan extraer. Permitiendo así obtener la información de las etiquetas de privacidad, que detalla los datos recopilados por la aplicación, los metadatos principales como los permisos requeridos, y las URLs de las políticas de privacidad.

ii. **Primer módulo (Descarga de los textos de políticas de privacidad):** Para este proceso, es necesario contar con la información de las aplicaciones móviles en

formato de conjunto de datos, así como las URLs de las políticas de privacidad obtenidas del Módulo UPM. Con esta información, se procede a descargar y asociar los archivos de políticas de privacidad con las aplicaciones móviles. Como resultado, se generan dos salidas: los archivos de políticas de privacidad y un conjunto de datos que incluye tanto la información de las aplicaciones como la de las políticas de privacidad.

iii. **Segundo módulo (Identificador de datos personales):** Este módulo utiliza como entrada las etiquetas de privacidad obtenidas del Módulo UPM y una lista de datos personales. Con esta información, se verifica si los datos recopilados por las aplicaciones móviles coinciden con los datos personales definidos. Esto permite determinar si la aplicación recolecta o no datos personales, y el resultado de esta identificación se guarda en un conjunto de datos resultante.

iv. **Tercer módulo (Identificador de permisos sensibles):** Este proceso utiliza los metadatos del Módulo UPM junto con una lista de permisos sensibles. Se compara la lista de permisos de las aplicaciones con los permisos sensibles definidos para identificar si las aplicaciones requieren permisos sensibles. Como resultado, se genera un conjunto de datos que refleja esta información.

v. **Cuarto módulo (Verificador de políticas de privacidad mediante IA):** Este módulo utiliza como entrada los resultados del primer módulo. Su objetivo es analizar, mediante inteligencia artificial, los textos de las políticas de privacidad para determinar si realmente lo son. Los resultados de este análisis se combinan con el conjunto de datos de las aplicaciones, centralizando toda la información en un único lugar.

vi. **Quinto módulo (Evaluación de cumplimiento):** Para el último módulo, se requiere como entrada los conjuntos de datos generados por el segundo, tercer y cuarto módulo, que contienen información sobre datos personales, permisos sensibles y políticas de privacidad, respectivamente. Con esta información, se evalúa el potencial cumplimiento de las aplicaciones en cuanto a la difusión de políticas de protección de datos. Los resultados se almacenan en un conjunto de datos que reúne toda la información relevante.

En la Figura 2.3, se presenta el diagrama de secuencia que ilustra la comunicación entre los módulos antes descrito.

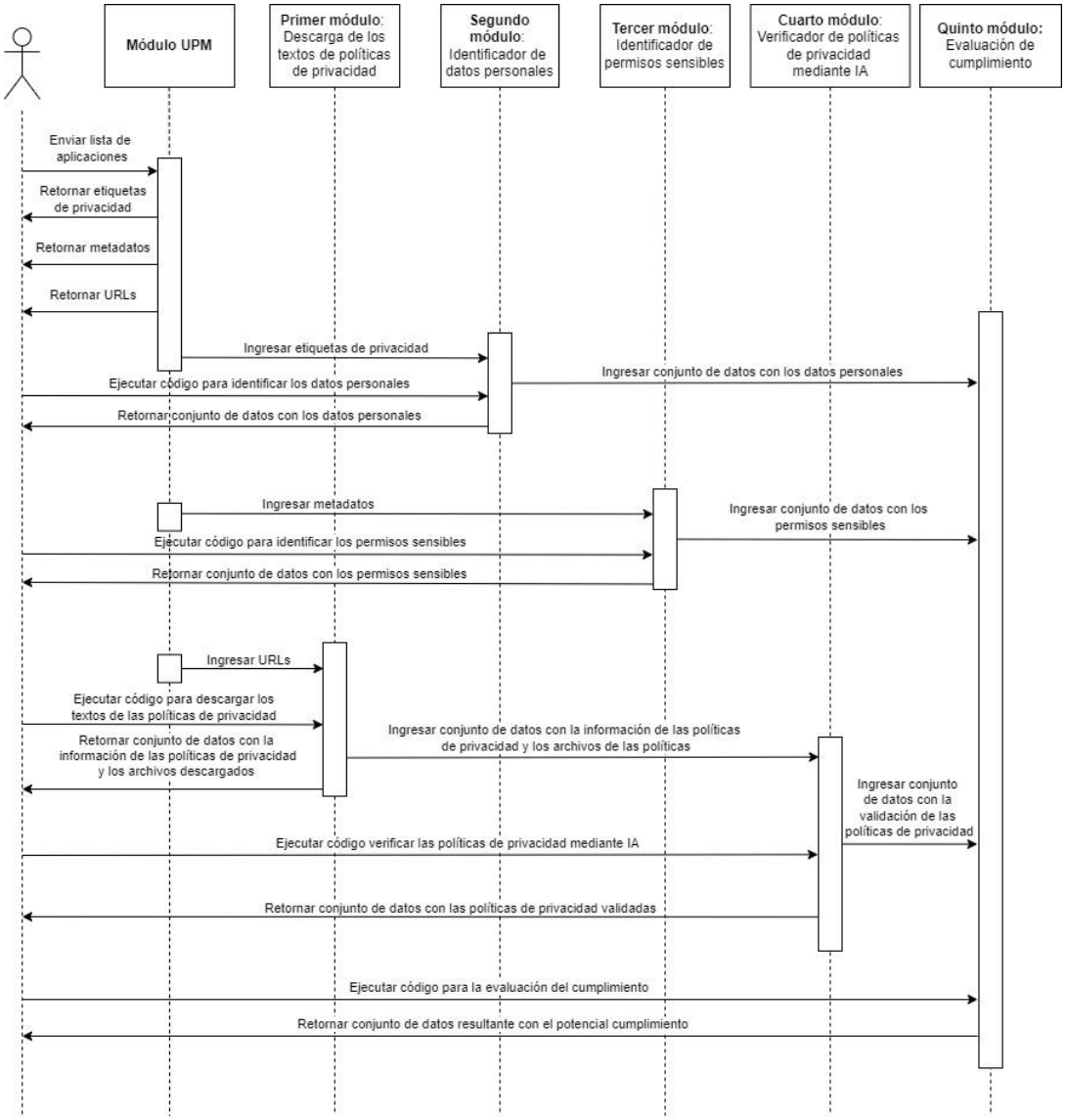

Figura 2.3 Interacción entre módulos para descarga y extracción de datos para evaluación de cumplimiento

2.2.1 Diseño del primer módulo: Descarga de los textos de políticas de privacidad

El primer módulo del proyecto utiliza un microservicio desarrollado previamente por J. Miranda [15]. El cual permite la descarga de las políticas de privacidad en formato de texto y como código HTML. A continuación, se presenta la arquitectura de este módulo en conjunto con el microservicio, así como el diagrama de actividades que muestra su comportamiento.

2.2.1.1 Arquitectura del primer módulo

En la Figura 2.4, se ilustra la arquitectura de este primer módulo.

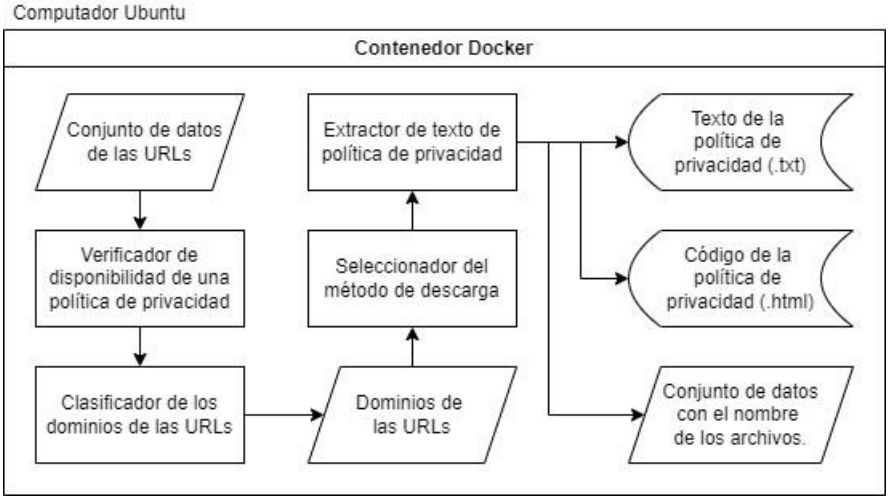

Figura 2.4 Arquitectura del primer módulo.

Los componentes de la arquitectura del primer módulo son los siguientes [15]:

i. **Conjunto de datos de las URLs:** Archivo en formato Excel que contiene información acerca de las aplicaciones móviles, donde se destaca: el nombre de la aplicación móvil y la dirección URL.
ii. **Verificador de disponibilidad de una política de privacidad:** Se verifica si la política de privacidad está en línea y está disponible para su descarga.
iii. **Clasificador de los dominios de las URLs:** Permite identificar el sitio web o el formato del documento de la política de privacidad.
iv. **Dominios de las URLs:** Se obtiene el dato acerca del proveedor del sitio web que contiene la política de privacidad.
v. **Seleccionador del método de descarga:** Se basa en el dominio de la URL para encontrar el mejor método entre los dominios considerados en la solución.
vi. **Extractor del texto de política de privacidad:** Realiza la extracción de la política de privacidad, guardando la información en formato de texto (.txt) y en código (.html), además genera un archivo Excel resultante que contendrá la información del nombre de la política de privacidad.

2.2.1.2 Diagrama de actividades del primer módulo

En la Figura 2.5, se presenta el diagrama de actividades el primer módulo.

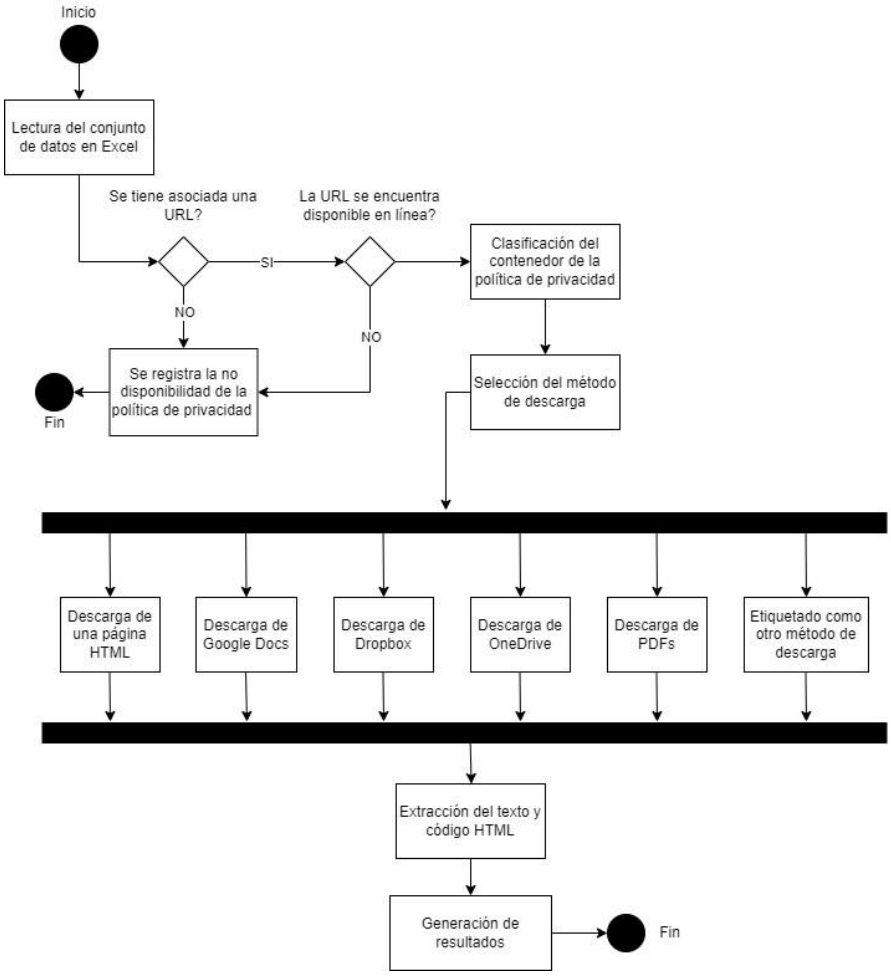

Figura 2.5 Diagrama de actividades del primer módulo.

A continuación, se realiza una breve descripción acerca de los procesos involucrados en el primer módulo [15]:

i. Lectura del conjunto de datos en Excel

Este proceso requiere de un conjunto de datos en Excel que contenga la información de las aplicaciones, entre las que se destacan las columnas del nombre de la aplicación y la dirección de URL de la política de privacidad.

ii. Verificación de que se esté asociada una URL

En este proceso se verifica si las aplicaciones tienen una URL de política de privacidad asociada. Esto se debe porque no en todas las aplicaciones se puede encontrar una política de privacidad o no se han podido obtener las URLs.

iii. Verificación la disponibilidad de la política de privacidad

Este proceso verifica mediante una petición HTTP get la disponibilidad de la política de privacidad.

iv. Clasificación del contenedor de una política de privacidad

El principal objetivo de este proceso es clasificar el contenedor de las políticas de privacidad, es decir, saber los dominios y subdominios de las políticas de privacidad. Los dominios y subdominios se irán relacionando con los dominios de los sitios web que han sido contemplados para la descarga de los documentos que contienen las políticas de privacidad.

v. Selección del método de descarga

En este proceso se selecciona el método más adecuado para descargar una política de privacidad. Para lograr esto, se requiere de la clasificación del contenedor de la política de privacidad. Se puede utilizar los siguientes métodos de descarga [15]:

- **Descarga de una página HTML. -** Este método permite la descarga de las políticas de privacidad que se encuentren en páginas HTML estáticas y dinámicas mediante *Selenium.* Para páginas dinámicas se realiza una espera de 30 a 60 segundos para garantizar que el código se haya renderizado.
- **Descarga de Google Docs. -** Método de descarga para documentos que se encuentran en Google Docs. Se descarga la política de privacidad mediante `requests`.
- **Descarga de One Drive. -** Este método de descarga permite obtener las políticas de privacidad que se encuentren dentro de One Drive y estén en formato DOCX, mediante `requests`. El documento de la política de privacidad es descargado en formato PDF.
- **Descarga de Dropbox. -** Se utilizan `requests` para descargar las políticas de privacidad que se encuentren en el sitio web Dropbox y con formatos como TXT, HTML, RTF, DOC, DOCX Y PDF. La descarga de las políticas de privacidad es en formato PDF para los formatos de RTF, DOC, DOCX y PDF. En cambio, para documentos con formato de TXT y HTML, la URL de la política contiene otra página de una política de privacidad de contenido estático [15].
- **Descarga de PDFs. -** Este método permite descargar documentos que se encuentren en formato PDF, esto se realiza mediante la configuración de ciertas opciones en la descarga de `requests` para así poder fijar el tamaño de flujo de descarga y la certificación de autenticidad de la página de la política de privacidad.

vi. Etiquetado con otro método de descarga

Este método sirve para descargas parciales, incompletas, vacías o de acceso denegado.

vii. Extracción del texto y código HTML

Este proceso es el que permite extraer la información de la política de privacidad dependiendo del método de descarga que se haya asociado a cada política de privacidad.

viii. Generar resultados

En este último proceso se generan los resultados de los archivos tanto en formato TXT como en HTML, ambos archivos se guardan con un formato predefinido de nombre combinando el nombre de paquete de Google y la fecha en la que se está generando el archivo. Adicionalmente, se genera un archivo de Excel que combina el conjunto de datos de entrada y el nombre del archivo de la política de privacidad.

2.2.2 Diseño del segundo módulo: Identificador de datos personales

Este módulo es el que permite identificar si las aplicaciones móviles están recolectando datos personales. A continuación, se presenta la arquitectura del módulo y el diagrama de actividades para explicar su funcionamiento.

2.2.2.1 Arquitectura del segundo módulo

En la Figura 2.6, se ilustra la arquitectura de este segundo módulo.

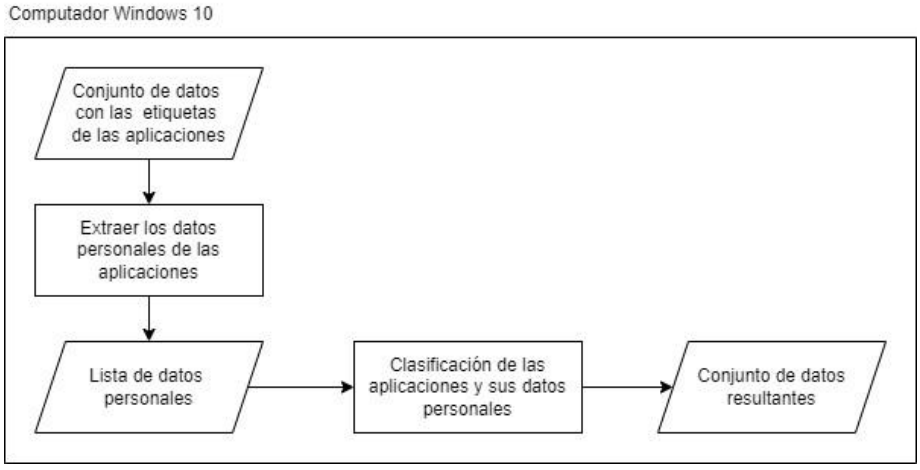

Figura 2.6 Arquitectura del segundo módulo.

A continuación, se detallan los elementos de la arquitectura:

i. **Conjunto de datos de las etiquetas de las aplicaciones:** Conjunto de datos descargados desde Play store que contiene los datos recolectados por las aplicaciones móviles.
ii. **Extraer los datos personales de las aplicaciones:** Proceso en el cual se extraen los datos personales que recolectaría cada aplicación (según la etiqueta de privacidad) y se los va a agrupar para su mejor manejo de datos.
iii. **Lista de datos personales:** Lista que contiene un conjunto base de categorías de datos personales. Esta lista se presenta a continuación:
    a. Dirección
    b. Ubicación aproximada
    c. Contactos
    d. Score de crédito
    e. Dirección de email
    f. Emails
    g. Archivos y documentos
    h. Información de estado físico
    i. Información de salud
    j. Nombre
    k. Otra información financiera
    l. Número de teléfono
    m. Fotografías
    n. Ubicación precisa
    o. Raza y etnia
    p. Orientación sexual
    q. SMS o MMS
    r. Información de pago del usuario
    s. Videos
iv. **Clasificación de las aplicaciones y sus datos personales:** Proceso en el que se verifican los datos personales declarados en las etiquetas de privacidad de las aplicaciones y se comparan con los de la lista definida en el paso anterior. El producto resultante es un conjunto de datos en Excel con la información del nombre de la aplicación y si recolecta o no datos personales, según la etiqueta de privacidad descargada desde Play Store.

2.2.2.2 Diagrama de actividades del segundo módulo

En la Figura 2.7, se presenta el diagrama de actividades del segundo módulo.

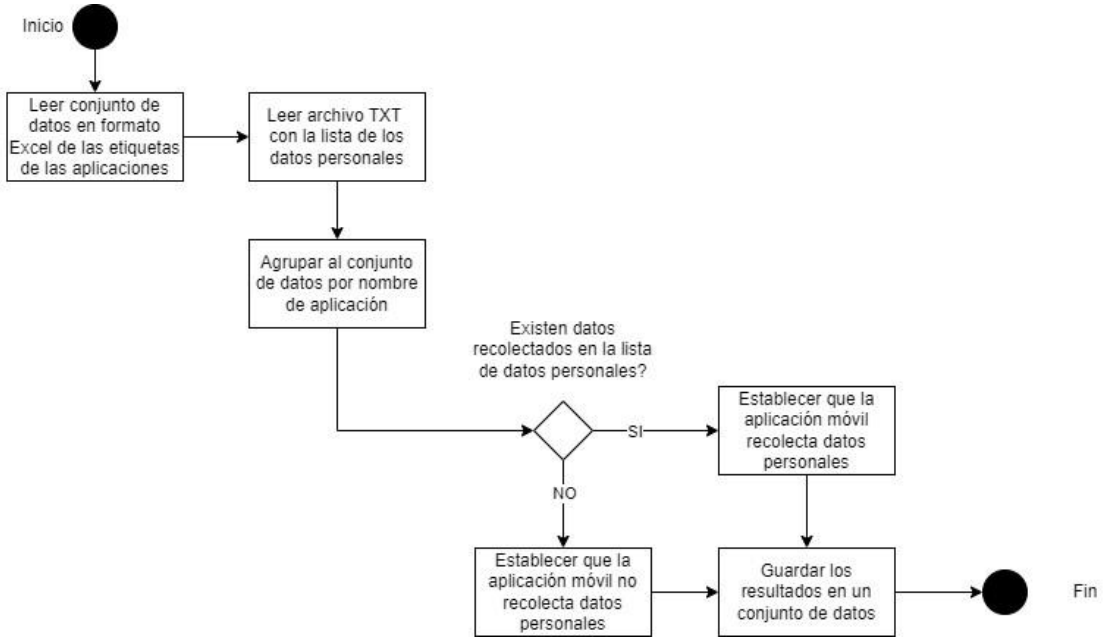

Figura 2.7 Diagrama de actividades del segundo módulo.

A continuación, se realiza una descripción acerca de los procesos involucrados en el segundo módulo:

i. Leer conjunto de datos en formato Excel de las etiquetas de las aplicaciones

En este proceso se lee el archivo Excel que contiene el conjunto de datos de las etiquetas de las aplicaciones, donde lo más importante es la información de los datos que cada aplicación recolecta y procesa.

ii. Leer archivo TXT con la lista de los datos personales

Para este proceso se realiza una lectura de un archivo TXT que contiene una lista base de categorías de datos personales y que servirán después para saber si efectivamente una aplicación recolecta este tipo de datos.

iii. Agrupar al conjunto de datos por nombre de aplicación

El objetivo de este proceso es simplificar la comparación entre los datos recolectados y procesados por la aplicación y los datos personales. Para lograrlo, es necesario agrupar estos datos y establecer un orden claro sobre cómo las aplicaciones procesan la información.

iv. Verificar si existen datos recolectados en la lista de datos personales

En este punto, se compara la información procesada por las aplicaciones con los datos listados. Si al menos uno de los datos procesados por la aplicación coincide con los de la lista, se considera que la aplicación recolecta y procesa datos personales. En caso contrario, si no hay coincidencias, se determina que la aplicación no recolecta ni procesa datos personales.

v. Establecer que la aplicación recolecta o no datos personales

Se establecen los siguientes valores de acuerdo con si la aplicación recolecta o no datos personales. Para el caso positivo, se define el valor como ‘SI’, caso contrario, ‘NO’.

vi. Guardar los resultados en un conjunto de datos

En este proceso final se genera un archivo en Excel resultante que contiene los nombres de las aplicaciones y una columna que especifica si se recolecta datos personales y cuyos valores son los mencionados previamente.

2.2.3 Diseño del tercer módulo: Identificador de permisos sensibles

Este módulo identifica si las aplicaciones móviles están solicitan permisos sensibles para acceder a datos sensibles desde el punto de vista de la privacidad. A continuación, se presenta la arquitectura del módulo y el diagrama de actividades para explicar su funcionamiento.

2.2.3.1 Arquitectura del tercer módulo

En la Figura 2.8, se presenta la arquitectura de este tercer módulo.

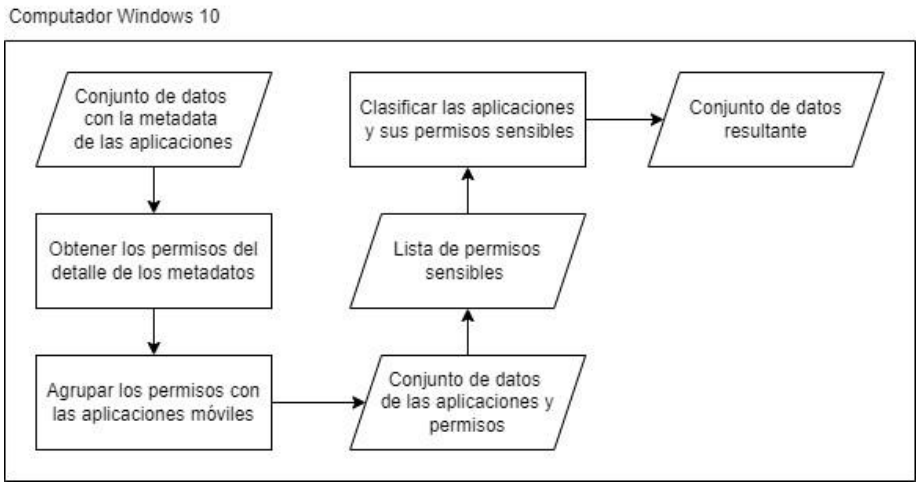

Figura 2.8 Arquitectura del tercer módulo.

A continuación, se realiza una descripción de cada elemento de la arquitectura del tercer módulo:

i. **Conjunto de datos con los metadatos de las aplicaciones:** Se tiene como entrada un conjunto de datos que tiene como información importante el nombre de la aplicación y los detalles de los metadatos, que fueron extraídos desde Play Store.
ii. **Obtener los permisos del detalle de los metadatos:** Se extraen los permisos de desde los metadatos de las aplicaciones móviles, que están representados en formato JSON.
iii. **Agrupar los permisos con las aplicaciones móviles:** Después de extraer los permisos de las aplicaciones móviles, estos se organizan en un conjunto de datos junto con los nombres de las aplicaciones, facilitando así su manejo.
iv. **Lista de permisos sensibles:** Lista base de permisos sensibles y que podrían atentar contra la privacidad del usuario. A continuación, se muestra la lista de permisos sensibles:
    a. RECEIVE_SMS
    b. READ_SMS
    c. CALL_PHONE
    d. ANSWER_PHONE_CALLS
    e. CAMERA
    f. RECORD_AUDIO
    g. BODY_SENSORS
    h. ACCESS_COARSE_LOCATION
    i. ACCESS_FINE_LOCATION
    j. ACCESS_BACKGROUND_LOCATION
    k. ACCESS_LOCATION_EXTRA_COMMANDS
    l. ACCESS_MEDIA_LOCATION
    m. WRITE_EXTERNAL_STORAGE
    n. READ_EXTERNAL_STORAGE
    o. READ_CONTACTS
    p. WRITE_CONTACTS
    q. READ_CALL_LOG
    r. WRITE_CALL_LOG
    s. GET_ACCOUNTS
    t. USE_BIOMETRIC
    u. USE_FINGERPRINT

v. **Clasificar las aplicaciones y sus permisos sensibles:** Se compara la lista de permisos solicitados por las aplicaciones con la lista base de permisos sensibles para determinar si las aplicaciones móviles requieren permisos considerados como sensibles y, por lo tanto, deberían publicar una política de privacidad.

2.2.3.2 Diagrama de actividades del tercer módulo

En la Figura 2.9, se ilustra el diagrama de actividades del tercer módulo.

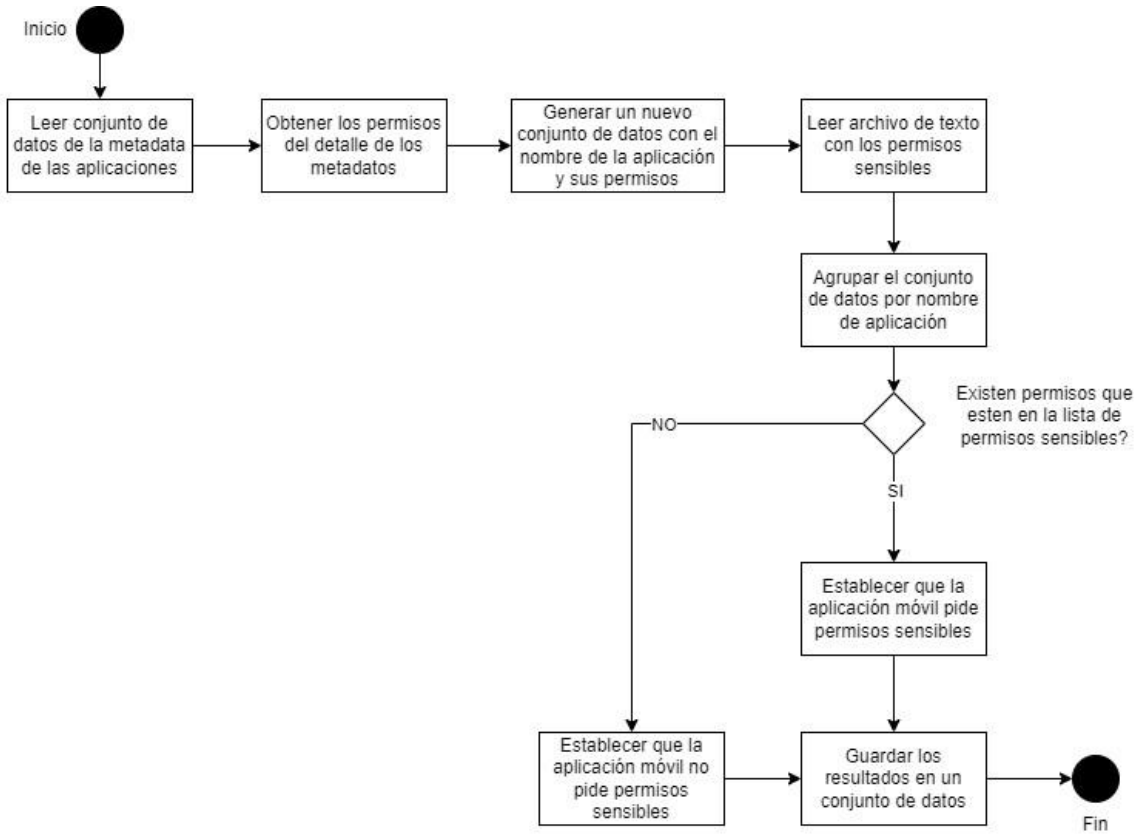

Figura 2.9 Diagrama de actividades del tercer módulo.

A continuación, se detallan los procesos involucrados en el diagrama:

i. Leer conjunto de datos de los metadatos de las aplicaciones

En este proceso, se lee un conjunto de datos que incluye el nombre de las aplicaciones móviles y sus metadatos, los cuales detallan los permisos que cada aplicación requiere del usuario.

ii. Obtener los permisos del detalle de los metadatos

Dado que los detalles de los metadatos están en formato JSON, es necesario extraer esta información en listas que luego podrán ser utilizadas.

iii. Generar un nuevo conjunto de datos con el nombre de la aplicación y sus permisos

Se asocian las listas extraídas de los metadatos con las aplicaciones móviles y se genera un nuevo conjunto de datos que incluirá estas dos columnas de información.

iv. Leer archivo de texto con los permisos sensibles

Se lee una lista de permisos considerados sensibles desde un archivo en formato TXT.

v. Agrupar el conjunto de datos por nombre de aplicación

En este paso, se agrupa el conjunto de datos según el nombre de la aplicación, lo que permite una visualización más clara de los permisos que cada aplicación recolecta.

vi. Verificar si los permisos de las aplicaciones móviles están en la lista de permisos sensibles

Se compara la lista de permisos requeridos por una aplicación con la lista de permisos sensibles para determinar si la aplicación solicita permisos considerados sensibles.

vii. Establecer si la aplicación móvil requiere o no de permisos sensibles

A cada aplicación móvil se le asigna un valor de 'SI' si requiere permisos sensibles y 'NO' en caso contrario.

viii. Guardar los resultados en un conjunto de datos

Se genera un conjunto de datos en el que se asocia el nombre de la aplicación móvil con una etiqueta de 'SI' o 'NO', indicando si la aplicación solicita permisos sensibles.

2.2.4 Diseño del cuarto módulo: Verificador de políticas de privacidad mediante IA

Este módulo permite, usando la Inteligencia Artificial (IA), analizar los textos descargados mediante las URLs publicadas en Play Store y verificar sí en efecto se tratan de políticas de privacidad. A continuación, se presenta la arquitectura del módulo y el diagrama de actividades explicar su funcionamiento.

2.2.4.1 Arquitectura del cuarto módulo

En la Figura 2.10, se presenta la arquitectura del cuarto módulo.

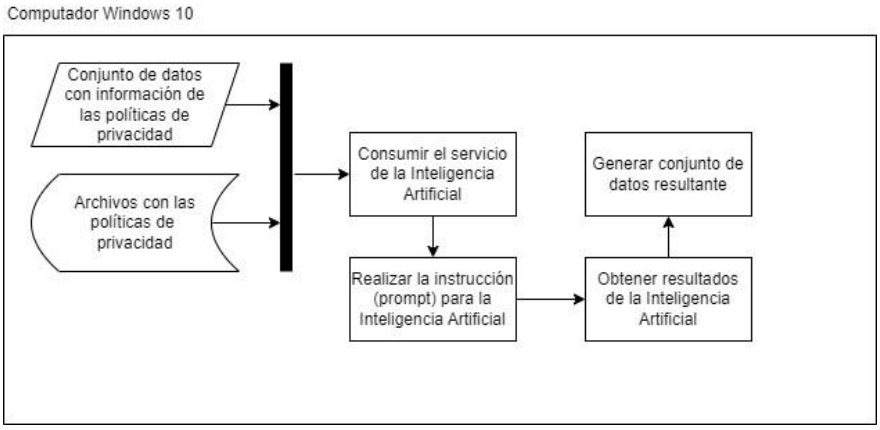

Figura 2.10 Arquitectura del cuarto módulo.

A continuación, se detallan los elementos de la arquitectura del cuarto módulo:

i. **Conjunto de datos con información de las políticas de privacidad:** Este elemento contiene el conjunto de datos con la información de las aplicaciones y los nombres de las políticas de privacidad asociadas. Estos nombres corresponden a los archivos que eventualmente contienen los textos de las políticas de privacidad.
ii. **Archivos con las políticas de privacidad:** Archivos de las políticas de privacidad en formato TXT y HTML.
iii. **Consumir el servicio de la Inteligencia Artificial:** Se utiliza un servicio de inteligencia artificial para ejecutar instrucciones mediante código Python.
iv. **Realizar la instrucción para la Inteligencia Artificial:** En este paso, se genera una instrucción para enviar a la inteligencia artificial, que incluye el texto de la política de privacidad y la pregunta sobre si el texto, efectivamente, corresponde a una política de privacidad.
v. **Obtener resultados de la Inteligencia Artificial:** Al enviar la instrucción a la inteligencia artificial, se espera una respuesta de 'SI' si el texto es realmente una política de privacidad cumple, o 'NO' en caso contrario. La respuesta será exclusivamente 'SI' o 'NO'.
vi. **Generar conjunto de datos resultante:** Finalmente, se genera un conjunto de datos que incluye elementos clave como el nombre de la aplicación móvil y los resultados proporcionados por la inteligencia artificial, indicando si cada aplicación cumple o no con los requisitos de una política de privacidad.

2.2.4.2 Diagrama de actividades del cuarto módulo

En la Figura 2.11, se ilustra el diagrama de actividades del cuarto módulo.

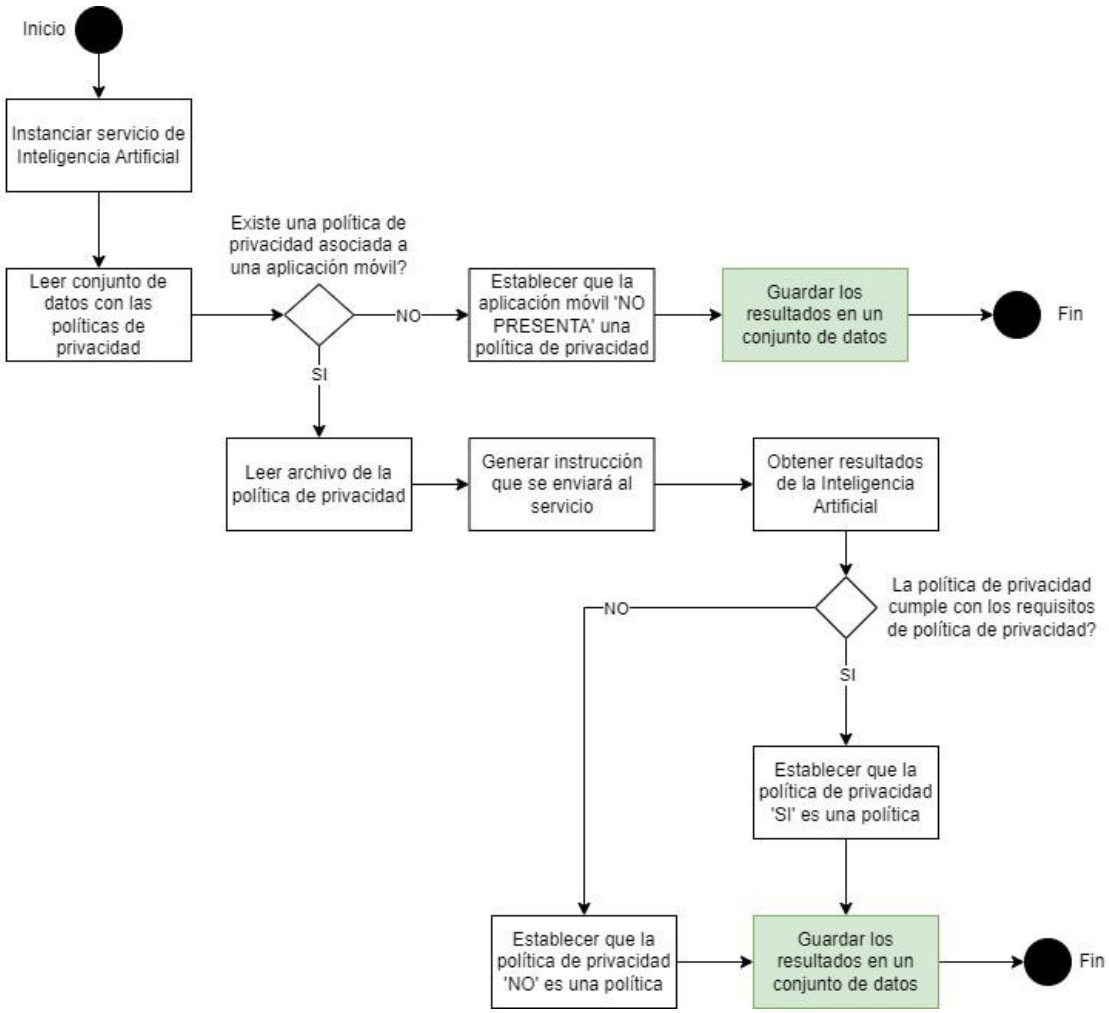

Figura 2.11 Diagrama de actividades del cuarto módulo.

A continuación, se detallan los procesos que se involucran en el cuarto módulo:

i. Instanciar servicio de Inteligencia Artificial

Este proceso permite configurar el servicio de inteligencia artificial, estableciendo la conexión necesaria para ejecutar las instrucciones de manera efectiva.

ii. Leer conjunto de datos con las políticas de privacidad

En esta actividad se lee el conjunto de datos que contiene la información de los nombres de las aplicaciones y el nombre de los archivos con los textos que eventualmente se tratan de políticas de privacidad.

iii. Verificar si la aplicación móvil tiene asociada una política de privacidad

Se verifica la existencia de una política de privacidad asociada a la aplicación móvil. Si la política no está presente, se etiqueta como 'NO PRESENTA'. En caso contrario, se continúa con el flujo de evaluación.

iv. Leer archivo de política de privacidad

Se lee el archivo de política de privacidad utilizando el nombre especificado en el conjunto de datos y se almacena en una variable.

v. Generar instrucción que se enviará al servicio

En este proceso, se genera la instrucción para la Inteligencia Artificial, en la que se define la tarea a realizar, se proporciona la política de privacidad y se especifica la salida esperada: 'SI' si el texto efectivamente se trata de una política de privacidad, o 'NO' en caso contrario.

vi. Obtener resultados de la Inteligencia Artificial

Se espera a que el servicio de Inteligencia Artificial responda la instrucción indicada donde los únicos valores posibles son ‘SI’ o ‘NO’

vii. Establecer salida para las políticas de privacidad

Se asocia a la aplicación móvil con la salida de la inteligencia artificial.

viii. Guardar los resultados en un conjunto de datos

Se guardan los resultados en un conjunto de datos, donde la información principal es el de los nombres de las aplicaciones y su etiqueta de si su política de privacidad cumple o no con los requisitos para una política de privacidad.

2.2.5 Diseño del quinto módulo: Evaluación de cumplimiento

En este módulo se realiza una clasificación sobre el potencial (in)cumplimiento de la Ley Orgánica de Protección de Datos, tomando en cuenta los conjuntos de datos obtenidos en los anteriores módulos. Este módulo presenta los resultados finales. A continuación, se presentará la arquitectura y el diagrama de actividades para esté módulo.

2.2.5.1 Arquitectura del quinto módulo

En la Figura 2.12, se muestra la arquitectura del quinto módulo.

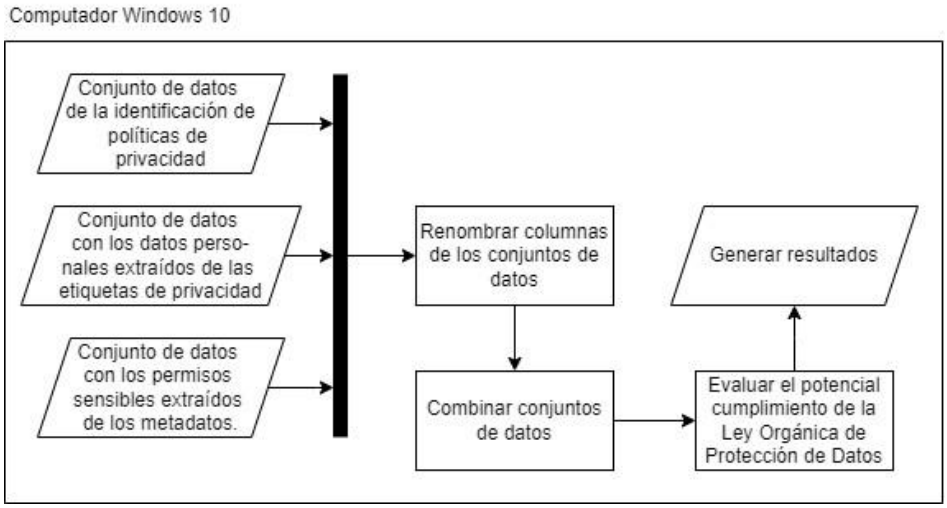

Figura 2.12 Arquitectura del quinto módulo.

A continuación, se presenta una descripción sobre los elementos de la arquitectura del quinto módulo:

i. **Conjunto de datos de la clasificación de políticas de privacidad:** Conjunto de datos con la información resultante del módulo cuatro, es decir, el nombre de las aplicaciones y la confirmación de que los textos descargados son realmente políticas de privacidad.
ii. **Conjunto de datos con los datos personales:** Conjunto de datos que contiene la información resultante del módulo dos, incluyendo el nombre de las aplicaciones e información que indica si recolectan o no datos personales (extraídos de las etiquetas)
iii. **Conjunto de datos con los permisos sensibles:** Conjunto de datos con la información resultante del módulo tres, donde se incluye la información de los nombres de las aplicaciones y la información que indica si las aplicaciones requieren o no de permisos sensibles.
iv. **Renombrar columnas de los conjuntos de datos:** Se renombran las columnas en cada conjunto de datos para asegurar que columnas similares tengan el mismo nombre en todos ellos. Por ejemplo, se unifica el nombre de la columna que contiene los nombres de las aplicaciones.
v. **Combinar conjuntos de datos:** Se combinan los tres conjuntos de datos para poder tener para cada aplicación la información pertinente de políticas de privacidad, datos personales y permisos sensibles.
vi. **Evaluar el potencial cumplimiento de la Ley Orgánica de Protección de Datos:** En este paso se evalúan las tres etiquetas: política de privacidad, datos personales

y permisos sensibles. Se revisan las combinaciones de estas etiquetas para determinar si, en conjunto, se cumple potencialmente con la Ley Orgánica de Protección de Datos Personales.

vii. **Generar resultados:** Se generan los resultados que indican el potencial cumplimiento de la Ley Orgánica de Protección de Datos Personales. Se crea un archivo con los conjuntos de datos que incluye toda la información relevante sobre las aplicaciones móviles, las etiquetas correspondientes y el potencial cumplimiento de la LOPDP.

2.2.5.2 Diagrama de actividades del quinto módulo

En la Figura 2.13, se ilustra el diagrama de actividades del quinto módulo.

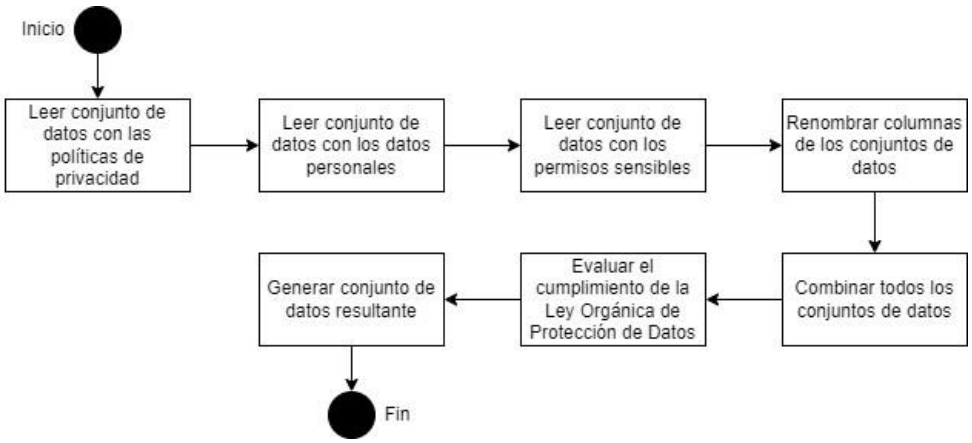

Figura 2.13 Diagrama de actividades del quinto módulo.

A continuación, se detalla los procesos involucrados en el quinto módulo:

i. Leer conjunto de datos con las políticas de privacidad

En este proceso se realiza la lectura del conjunto de datos que contiene la información de las políticas de privacidad.

ii. Leer conjunto de datos con los datos personales

De manera similar al paso anterior, se lee el conjunto de datos que contiene información sobre los datos personales recolectados por las aplicaciones móviles (declarados en las etiquetas).

iii. Leer conjunto de datos con los permisos sensibles

De manera similar a los pasos anteriores, en este se lee el conjunto de datos que detalla los permisos sensibles que las aplicaciones móviles están solicitando a los usuarios.

iv. Renombrar columnas de los conjuntos de datos

En este paso, se modifican los nombres de las columnas que contienen información común en los tres conjuntos de datos. Se unifica el nombre de la columna que contiene los nombres de las aplicaciones para facilitar la combinación de los conjuntos de datos.

v. Combinar todos los conjuntos de datos

Se combinan los tres conjuntos de datos para consolidar la información sobre políticas de privacidad, datos personales y permisos sensibles en un solo archivo. Esto facilita el análisis del cumplimiento de la Ley Orgánica de Protección de Datos Personales.

vi. Evaluar el cumplimiento de la Ley Orgánica de Protección de Datos

Se realiza la evaluación del cumplimiento de la Ley Orgánica de Protección de Datos basado en las etiquetas de políticas de privacidad, datos personales y permisos sensibles. Para esto se define la Tabla 2.1, la cual define la clasificación del potencial cumplimiento de la LOPDP.

Tabla 2.1 Clasificación del potencial cumplimiento de la LOPDP.

Recolecta datos personales	Requiere permisos sensibles	Política de privacidad	Potencial cumplimiento
NO	NO	NO PRESENTA	NO APLICA
NO	NO	NO	NO APLICA
NO	NO	SI	NO APLICA
NO	SI	NO PRESENTA	NO
NO	SI	NO	NO
NO	SI	SI	NO
SI	NO	NO PRESENTA	NO
SI	NO	NO	NO
SI	NO	SI	SI
SI	SI	NO PRESENTA	NO
SI	SI	NO	NO
SI	SI	SI	SI

A continuación, se define los criterios de la clasificación:

Cumple (SI):

- La aplicación declara en la etiqueta de privacidad que recolecta datos personales, no pide permisos sensibles, y presenta una política de privacidad válida.
- La aplicación declara en la etiqueta de privacidad que recolecta datos personales, pide permisos sensibles, y presenta una política de privacidad válida.

No Cumple (NO):

- La aplicación declara en la etiqueta de privacidad que no recolecta datos personales, pero pide permisos sensibles (independientemente si presenta o no una política de privacidad) se considera como un posible incumplimiento. Esto debido a que existe una incongruencia entre la declaración en la etiqueta de privacidad y que pidan permisos sensibles.
- La aplicación no presenta una política de privacidad, aun cuando sí recolecta datos personales o pide permisos sensibles.

No Aplica (NO APLICA):

- La aplicación no recolecta datos personales y no pide permisos sensibles, pero presenta una política de privacidad.
- La aplicación no recolecta datos personales y no pide permisos sensibles, y no presenta una política de privacidad. En este caso, se considera que no es necesario llevar el cumplimiento.

vii. Generar conjunto de datos resultante

Se genera un conjunto de datos resultante que contiene esta clasificación del potencial cumplimiento de la LOPDP. El archivo resultante será de formato Excel para analizar de manera más sencilla estos resultados.

2.3 Implementación

A continuación, se presenta la implementación de los distintos módulos, incluyendo las secciones de código más relevantes.

2.3.1 Implementación del Módulo UPM

Se solicitó asistencia a la Universidad Politécnica de Madrid (UPM) debido a que dispone de un software especializado para la extracción de datos relevantes de aplicaciones móviles. En consecuencia, se envió a esta universidad la lista de aplicaciones previamente seleccionadas y los datos necesarios para el análisis de los criterios de evaluación. Esta lista, compuesta por un total de 379 aplicaciones, se encuentra en el ANEXO C.

Como resultado de esta colaboración, se obtuvieron las etiquetas de privacidad de las aplicaciones móviles, que incluyen información clave sobre los datos recolectados y los enlaces (URLs) a las políticas de privacidad correspondientes. También se recopiló

información sobre los metadatos de las aplicaciones, con un enfoque particular en los permisos solicitados a los usuarios.

Además, UPM proporcionó los archivos de las políticas de privacidad en formato de texto. Estos archivos serán comparados con los obtenidos en la implementación del primer módulo. La comparación se llevará a cabo en la subsección de implementación del cuarto módulo para evaluar cuáles políticas de privacidad son las más adecuadas.

2.3.2 Implementación del primer módulo

En esta sección se describen los pasos necesarios para la correcta implementación del primer módulo, que incluye la creación y ejecución de una imagen Docker. Se analiza el código, destacando la clasificación de las URLs y la selección del método de descarga más adecuado para obtener las políticas de privacidad.

En este módulo, se detallan las funciones que permiten una descarga exitosa a través de diversas tecnologías web. Además, se proporciona una explicación sobre la implementación del código que facilita la descarga de las políticas de privacidad en formatos TXT y HTML [15]. Finalmente, se presentan detalles sobre el archivo resultante que contendrá la información de las aplicaciones y el nombre de las políticas de privacidad.

En la Figura 2.14 del ANEXO D, se presenta el diagrama de despliegue del microservicio que se encuentra en el primer módulo. Para la explicación del código es necesario utilizar el diagrama de actividades que se encuentra en la Figura 2.5.

La lectura del archivo que contiene el conjunto de datos de las aplicaciones se realiza dentro de la función llamada '*Service3*'. Esta función lee el archivo '*aplicaciones_url.xlsx*', que debe incluir obligatoriamente las siguientes columnas: '*nombre_app', 'categoria', 'numero_descargas', 'package_name', 'ubicacion'* y '*privacy_policy_url*'. Esta estructura de datos se muestra en la Figura 2.15.

```
523  def Service3():
524      print('Entering microservice 3')
525      path = 'aplicaciones_url.xlsx'
526      elements = apk_list(path)
527      cont = 0
528      for index, row in elements.iterrows():
529          nombreApp = row['nombre_app']
530          categoria = row['categoria']
531          numeroDescargas = row['numero_descargas']
532          packageName = row['package_name']
533          ubicacion = row['ubicacion']
534          url = row['privacy_policy_url']
```

Figura 2.15 Lectura de archivo e instancia de columnas.

En la Figura 2.16, se muestra la validación de la URL asociada a la aplicación móvil mediante la función '*isNullorEmpty()*'. Si se encuentra una URL de política de privacidad, el flujo de procesamiento continúa normalmente. En caso contrario, se muestra un mensaje en la consola indicando que no se ha encontrado la URL, y se guarda la información de la aplicación con los campos de políticas de privacidad vacíos.

```
if isNullorEmpty(url) == False:
    [state, code] = get_status_code(url)
    logger.info(str(state) + ' , ' + str(code))
    if state == True and code == 200:
    else:
            logger.info('Problems with the URL response')
            print('Problems with the url state')
            doc_results.append((nombreApp, categoria, numeroDescargas, packageName, ubicacion, url, None))
else:
    print(f"No URL found in {packageName}")
    doc_results.append((nombreApp, categoria, numeroDescargas, packageName, ubicacion, None, None))
```

Figura 2.16 Validación de existencia de URL de política de privacidad.

El siguiente paso es verificar la disponibilidad de la política de privacidad. Para ello, se utiliza el método '*get_status_code(url)*', en el cual el parámetro de entrada es la URL de la política de privacidad. El método principal de esta función es '*urllib.request.urlopen().read()*' (línea 166), que se emplea para obtener la respuesta del servidor que aloja la política de privacidad. Si la solicitud se realiza correctamente y no se produce ninguna excepción, se devuelve un código HTTP 200 y se retorna el valor '*True*' en esta función, como se muestra en la Figura 2.17 [15].

```
161 def get_status_code(url):
162     headers = {'User-Agent':
163     'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 '
164     '(KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36'}
165     try:
166         req = urllib.request.Request(url, data=None, headers=headers)
167         urllib.request.urlopen(req, timeout=120)
168     except urllib.error.HTTPError as e:
169         if e.code == 404:
170             reason = 'Privacy policy unavailable'
171             logger.error("Privacy policy download failed",
172                          extra={'exception_message': str(e),
173                                 'reason': reason, 'exit_code': e.code})
174             return False, e.code
175         else:
176             return True, e.code
177     except urllib.error.URLError as e:
178         reason = 'Cannot connect to the domain server'
179         logger.error("Privacy policy download failed",
180                      extra={'exception_message': str(e),
181                             'reason': reason, 'url': url})
182         return False, e.reason
183     except Exception as e:
184         reason = 'Timeout in urllib.request.urlopen'
185         logger.error("Privacy policy download failed", extra={
186             'exception_message': str(e), 'reason': reason, 'url': url})
187         return False, str(e)
188     else:
189         return True, 200
```

Figura 2.17 Función para obtener la disponibilidad de la URL de la política de privacidad

En la Figura 2.18, se muestra la función encargada de seleccionar el método de descarga según el nombre del sitio web, basado en el dominio o subdominio de la URL de la política de privacidad.

```
112     # Function to determine the membership of the url to a website
113     def url_selector(url):
114         try:
115             csf_html = False
116             # To download pdf from common web page
117             csf_pdf = is_pdf_web(url)
118             # To download the docs format document from google docs website
119             token_docs = ['docs']
120             csf_docs = url_matching(url, token_docs)
121             # To download the txt format file from drive website
122             token_drive = ['drive']
123             csf_drive = url_matching(url, token_drive)
124             # To download the txt, html, doc, docx and pdf format file from drive website
125             token_dropbox = ['dropbox']
126             csf_dropbox = url_matching(url, token_dropbox)
127             # To download the docx format document from onedrive website
128             token_onedrive = ['onedrive', 'live']
129             csf_onedrive = url_matching(url, token_onedrive)
130             # if it is not any of the special cases, it is considered a web page of
131             if csf_pdf == False and csf_onedrive == False and \
132                     csf_drive == False and csf_dropbox == False and csf_docs == False:
133                 logger.info('Because it does not match any of the website domains under '
134                             'consideration, it is categorized by default as an HTML page.')
135                 csf_html = True
136         except Exception as e:
137             reason = 'url_selector unavailable'
138             logger.error('url_selector failed',
139                          extra={'exception_message': str(e), 'reason': reason})
140         else:
141             logger.debug('The function was successful')
142             return csf_pdf, csf_docs, csf_drive, csf_html, csf_dropbox, csf_onedrive
```

Figura 2.18 Función para la selección del método de descarga adecuado.

A continuación, se describen las funciones de los métodos de descarga. El primero, denominado '*download_general_text()*', se utiliza principalmente para descargar una página HTML mediante *Selenium*. Esta herramienta permite extraer el contenido de una política de privacidad desde una página web común. Para abordar el intercambio de peticiones entre el servidor y el navegador, se implementó un temporizador de 30 a 60 segundos. Esto ayuda a evitar problemas como la descarga parcial del contenido y permite que JavaScript modifique el contenido renderizado en la página web. Tal y como se observa en la Figura 2.19 [15].

```
191     def download_general_text(url):
192         policy_text = None
193         policy_html = None
194         TIMEOUT = 60
195         TIMERSLEEP = 30
```

Figura 2.19 Fragmento de código con la implementación de un temporizador.

En la Figura 2.20, se presentan las configuraciones del Webdriver, las cuales se detallan, a continuación [15]:

- `--no-sandbox`: Desactiva el sandbox para todos los procesos, útil en pruebas para omitir las medidas de seguridad del sistema operativo.
- `--enable-javascript`: Habilita la ejecución de JavaScript
- `--headless`: Inicia el navegador web sin interfaz gráfica.
- `--disable-dev-shm-usage`: Deshabilita el uso de la partición /dev/shm, que en algunos entornos virtuales puede ser demasiado pequeña.

```
198     chromeOptions.add_argument("--no-sandbox")
199     chromeOptions.add_argument("--enable-javascript")
200     chromeOptions.add_argument("--headless")
201     chromeOptions.add_argument('--disable-dev-shm-usage')
```

Figura 2.20 Fragmento de código con la configuración del Webdriver.

En la Figura 2.21, se muestra que, para extraer texto y código de la política de privacidad, es esencial conocer la estructura del Modelo de Objeto de Documento (DOM, por sus siglas en inglés). El DOM es un conjunto de herramientas diseñadas para manejar XML y, en particular, HTML. Tras identificar la etiqueta *<html>* en la línea 215, se puede utilizar el atributo *innerText* para recuperar y establecer el contenido de la política de privacidad como texto sin formato [15].

```
204     # Set the options
205     driver = webdriver.Chrome(executable_path=r'{}'
206                               .format(chromedriver_path), options=chromeOptions)
207     try:
208         logger.debug('The webdriver was being started')
209         WebDriverWait(driver, TIMEOUT).until(EC.presence_of_element_located(
210             (By.TAG_NAME, "html")))
211         # Get the HTML code from the page
212         driver.get(url)
213         time.sleep(TIMERSLEEP)
214         # Get the HTML code from the page
215         element = driver.find_element_by_tag_name('html')
216         # Extract text from the attribute innerText
217         policy_text = element.get_attribute('innerText')
```

Figura 2.21 Fragmento de código que extrae el texto y código de una política de privacidad

Como el título extraído de cualquier método de descarga se basa inicialmente en el nombre de la política de privacidad, se decidió generalizar el proceso de guardar estos títulos. En la Figura 2.22, se muestra la implementación de la función '*fix_title()*', que recibe como parámetro el nombre del paquete de la aplicación y lo combina con un patrón de

fecha y hora actual. Esto facilita la identificación de las políticas de privacidad para cada aplicación móvil.

```
463     def fix_title(title):
464         dateNow = datetime.now()
465         date_hour = dateNow.strftime('%m%y%H%M')
466         titleFixed = title+'_'+date_hour
467         return titleFixed
```

Figura 2.22 Función que permite dar formato al nombre de las políticas de privacidad.

Una vez generado el archivo y dado formato al título de las políticas de privacidad. En la Figura 2.23, se presenta la función denominada '*store_text()*' la cual recibe los siguientes argumentos de entrada [15]:

- *policytxt*: El cual contiene el texto de la política de privacidad.
- *policyhtml*: El cual contiene el código HTML de la política de privacidad.
- *title*: Título del archivo de la política de privacidad.

```
469     def store_text(policytxt, policyhtml, title):
470         try:
471             logger.debug('store_text function has been started')
472             file = open(result_dir + title + ".txt", "w")
473             file.write(policytxt)
474             file.close()
475             file = open(result_dir + title + ".html", "w")
476             file.write(policyhtml)
477             file.close()
478         except Exception as e:
479             reason = 'store_text function unviable'
480             logger.error("store_text failed",
481                          extra={'exception_message': str(e), 'reason': reason})
482         else:
483             logger.info('store_text function has been successful')
```

Figura 2.23 Función para el almacenamiento de la política de privacidad.

El siguiente método de descarga se basa en Google Docs, que contiene documentos web con las políticas de privacidad en formato DOCX. Para obtener estas políticas, se utiliza el código mostrado en la Figura 2.24, que emplea el método '*soup.find_all()*'. Este método busca etiquetas *<script>* del tipo *text/javascript*. Una vez localizadas, se extrae el texto completo utilizando una expresión regular con el método '*re.findall()*' [15].

```
248         js_text_lst = soup.find_all('script', type='text/javascript')
249         for js_text in js_text_lst:
250             js_text = str(js_text)
251             # Splitting and filtering the text matching with [XXXXX].
252             for text in re.findall("\[.+\]", js_text):
```

Figura 2.24 Fragmento de código para la selección de cadenas de caracteres contenidas en Google Docs.

En la Figura 2.25, se verifica que las cadenas de caracteres cumplan con el patrón *"ty":"is"*. Si el patrón es encontrado, se procede a realizar reemplazos en ciertas palabras del texto. Luego, se evalúa el texto extraído. La clave *'s'* es utilizada por Google Docs para identificar el texto relevante [15].

```
255                 if text is not None and '"ty":"is"' in text:
256                     text = text.replace('true', 'True')
257                     text = text.replace('false', 'False')
258                     text = text.replace('null', 'None')
259                     policy_text += ast.literal_eval(text)[0]['s']
```

Figura 2.25 Fragmento de código para la selección de texto en Google Docs.

El siguiente método de descarga está diseñado para obtener políticas de privacidad en formato PDF. Para ello, se utiliza la función '*download_pdf()*', que recibe como argumento la URL de la política de privacidad. La descarga se inicia mediante el método '*requests.get()*', configurado con *stream=True* para permitir la descarga en flujo y *verify=False* para omitir la verificación de certificados SSL. El archivo se guarda en modo *wb* (write bytes), con un tamaño de fragmento de 1024 bytes. Este procedimiento se detalla en la Figura 2.26 [15].

```
269 def download_pdf(url):
270     try:
271         logger.debug('download_pdf function has been started')
272         n_ram = random.randrange(10, 100, 4)
273         pdf_name = 'privacyPolicy' + str(n_ram)
274         response = requests.get(url, stream=True, verify=False)
275         file = open(result_dir + pdf_name + '.pdf', 'wb')
276         for chunk in response.iter_content(chunk_size=1024):
277             if chunk:
278                 file.write(chunk)
279     except Exception as e:
280         reason = 'Error while downloading pdf documento from the web'
281         logger.error("download_pdf download failed",
282                      extra={'exception_message': str(e), 'reason': reason})
283     else:
284         logger.info('ownload_pdf function has been started successful')
285         return pdf_name
286     finally:
287         file.close()
```

Figura 2.26 Función para descargar documentos PDF.

A continuación, en la Figura 2.27, se presenta la función '*pdf2text()*', que convierte un archivo PDF extrayendo su contenido mediante el método raw de la librería *Tika* y lo guarda en un archivo TXT.

```
412 def pdf2text(file_name):
413     try:
414         logger.debug('Text extraction from PDF document started')
415         raw = parser.from_file( result_dir + file_name+'.pdf')
416         content = raw['content']
417         file = open( result_dir + file_name + ".txt", "w")
418         file.write(content)
419         file.close()
420     except Exception as e:
421         reason = 'Error while extract text from pdf document'
422         logger.error("download_pdf download failed",
423                      extra={'exception_message': str(e), 'reason': reason})
424     else:
425         logger.debug('Extraction of text from a PDF document was successful')
```

Figura 2.27 Función para extraer texto de los documentos PDF.

En el sitio web de OneDrive, se pueden almacenar documentos de políticas de privacidad en formato DOCX. Para acceder a estos documentos, es necesario descargar el código HTML utilizando el método '*requests.get()*'. Posteriormente, se emplea este código HTML para localizar la URL del contenedor del texto de la política de privacidad mediante expresiones regulares. Además, se busca la línea que comience con la expresión '*Var $Config =*' y se identifica una etiqueta específica que contiene la URL de la política de privacidad. Este proceso se detalla en la Figura 2.28 [15].

```
304         tag = '"FileGetUrl"'
305         lines = policy_html.split('\n')
306         aux = [x for x in lines if x.startswith('var $Config=')]
307         url_text = aux[0].split(tag)[1].split('"')[1]
```

Figura 2.28 Fragmento de código para la búsqueda de URL del contenido de una política de privacidad.

Posteriormente, se descarga el documento de la política de privacidad directamente desde la URL que contiene la política de privacidad donde se va a establecer la extensión del archivo como DOCX. En la Figura 2.29, se observa cómo se extrae el documento DOCX que tiene una estructura XML usando la biblioteca *docx2txt* [15].

```
315         aux_response = requests.get(url_text, stream=True, verify=False)
316         file = open(result_dir + file_name + '.docx', 'wb')
317         for chunk in aux_response.iter_content(chunk_size=1024):
318             if chunk:
319                 file.write(chunk)
320         file.close()
321         policy_text = docx2txt.process(result_dir + file_name + '.docx')
```

Figura 2.29 Fragmento de código para la descarga y extracción de una política de privacidad.

Para el sitio web Dropbox se definió una función para obtener las políticas de privacidad las cuales están alojadas principalmente en formatos como: TXT, HTML, PDF, DOC y DOCX. La descarga del código HTML se lo realiza mediante el método '*requests.get()*'. En el código de la Figura 2.30, se presenta el fragmento de código que permite la selección del método de descarga de un documento que se encuentra alojado en Dropbox. Se utilizan condicionales para ejecutar funciones adecuadas dependiendo del formato del documento [15].

```
446             tag = '"preview_url"'
447             url_text = OD_filtrado(policyhtml, tag)
448             if formato == 'html':
449                 OD_html_store(file_name, url_text)
450             elif formato == 'txt':
451                 OD_text_store(file_name, url_text)
452             elif formato == 'docx' or formato == 'doc' or \
453                     formato == 'rtf'or formato == 'pdf':
454                 OD_pdf_store(file_name, url_text, url)
455                 pdf2text(file_name)
```

Figura 2.30 Fragmento de código que permite la selección del método de descarga en Dropbox

A continuación, en la Figura 2.31, se presenta el código de la función '*OD_filtrado()*', el cual recibe como argumento de entrada la variable *policyhtml*, que contiene el código HTML de las políticas de privacidad. La variable tag contiene el patrón que permite la búsqueda y extracción de la URL que contiene la política de privacidad [15].

```
373 def OD_filtrado(policyhtml, tag):
374     url_text = None
375     aux = None
376     try:
377         logger.debug('OD_filtrado function has been started')
378         lines = policyhtml.split('\n')
379         aux = [x for x in lines if x.startswith('InitReact.mountComponent')]
380         url_text = aux[0].split(tag)[1].split('"')[1]
381     except Exception as e:
382         reason = 'Error while filter url from dropbox'
383         logger.error("download_pdf download failed",
384                      extra={'exception_message': str(e), 'reason': reason})
385     else:
386         logger.debug('OD_filtrado function has been successful')
387         return url_text
```

Figura 2.31 Función para la extracción de la URL que contiene la política de privacidad.

Una vez que se pudo extraer la URL que contiene la política de privacidad se utiliza la función '*OD_text_store()*' para poder extraer el contenido de la política de privacidad y

almacenarlo en formato de texto TXT y código HTML, como se observa en la Figura 2.32 [15].

```
352     def OD_text_store(file_name, url_text):
353         try:
354             logger.debug('The download txt document from dropbox was start')
355             response = requests.get(url_text)
356             html = response.content
357             soup = BeautifulSoup(html, 'html.parser')
358             policy_html = (str(soup))
359             policy_text = soup.find('body').pre.text
360             file = open( result_dir + file_name + '.txt', "w")
361             file.write(policy_text)
362             file.close()
363             file = open( result_dir + file_name + '.html', "w")
364             file.write(policy_html)
365             file.close()
366         except Exception as e:
367             reason = 'Error while downloading txt documento from dropbox'
368             logger.error("download_pdf download failed",
369                          extra={'exception_message': str(e), 'reason': reason})
370         else:
371             logger.debug('The download txt document from dropbox was successful')
```

Figura 2.32 Fragmento de código que almacena los documentos de política de privacidad.

Posteriormente, en la Figura 2.33, se presenta la función '*OD_html_store()*', que recibe como argumentos el nombre del archivo y el texto obtenido de la URL de la política de privacidad. Esta función se encarga de almacenar el texto en el archivo correspondiente [15].

```
330     def OD_html_store(file_name, url_text):
331         try:
332             logger.debug('The download html document from dropbox was start')
333             response = requests.get(url_text)
334             html = response.content
335             soup = BeautifulSoup(html, 'html.parser')
336             policy_html = soup.find('body').pre.text
337             soup = BeautifulSoup(policy_html, 'html.parser')
338             policy_text = soup.find('body').text
339             file = open(result_dir + file_name + '.txt', "w")
340             file.write(policy_text)
341             file.close()
342             file = open(result_dir + file_name + '.html', "w")
343             file.write(policy_html)
344             file.close()
345         except Exception as e:
346             reason = 'Error while downloading html documento from dropbox'
347             logger.error("download_pdf download failed",
348                          extra={'exception_message': str(e), 'reason': reason})
349         else:
350             logger.debug('The download html documento from dropbox was successful')
```

Figura 2.33 Función que permite el almacenamiento de un documento HTML.

Para los documentos en formatos PDF, DOC, DOCX o RFT, se obtiene la URL del contenido de la política de privacidad, lo que permite realizar la descarga directa del

documento en el formato correspondiente. La descarga del contenido en formato PDF se ilustra en la Figura 2.26, mientras que la extracción del texto del documento se detalla en el código de la Figura 2.27 [15].

Para facilitar el manejo de los datos que se han ido obteniendo con respecto a las políticas de privacidad se guarda toda la información den un conjunto de datos en formato Excel (XLSX) añadiendo al conjunto de datos original la columna de '*policy_privacy_name*' y nombrando al conjunto de datos como '*doc_file.xlsx*'. Esta función lleva por nombre '`save_data()`', como se observa en la Figura 2.34.

```
495     # Save the data info into a csv file
496     def save_data(results):
497         try:
498             # Save the results into a Data Frame
499             results_df = pd.DataFrame(results, columns=['nombre_app','categoria',
500             'numero_descargas','package_name','ubicacion','policy_privacy_url',
501             'policy_privacy_name'])
502             # Save the results into a CSV
503             results_df.to_excel(doc_file, index=False)
504         except Exception as e:
505             reason = 'error while saving excel'
506             logger.error("save data into excel failed",
507                          extra={'exception_message':str(e),'reason': reason})
```

Figura 2.34 Función para guardar conjunto de datos resultante.

Adicionalmente, dado que UPM no pudo proporcionar automáticamente todos los enlaces de políticas de privacidad, es necesario buscar los enlaces faltantes de manera manual. De un total de 379 aplicaciones, se obtuvieron automáticamente 340 enlaces de políticas de privacidad. Sin embargo, tras filtrar manualmente, se descubre que algunas de las políticas descargadas contenían información mínima o nula. Como resultado, se realiza un conteo de las políticas que no se descargaron correctamente y se lleva a cabo una búsqueda manual para obtener 59 políticas adicionales.

Como se puede observar en la Figura 2.4, todo el primer módulo está basado en el microservicio el cual se encuentra dentro de un contenedor Docker, por lo tanto, es necesario crear la imagen Docker, la cual contendrá todo el entorno necesario para ejecutar la aplicación, en este caso el código Python.

En la Figura 2.35, se encuentra la primera parte del archivo de Docker.

```
#Select the SO
FROM ubuntu:18.04
# Dates of developer
LABEL mantenedor="zahid.copara@gmail.com"
LABEL version="1.0"
LABEL description="Imagen Microservicio"
# Local Variables
ENV LANG C.UTF-8
# libraries and dependences
RUN apt update && \
apt -y upgrade && \
apt install -y openjdk-11-jdk && \
apt install -y wget && \
apt install unzip && \
apt install -y git && \
apt install -y python3 && \
apt install -y python3-pip && \
pip3 install  python-json-logger && \
pip3 install python3-pika  && \
pip3 install tika && \
pip3 install selenium && \
pip3 install tld && \
pip3 install beautifulsoup4 && \
pip3 install docx2txt && \
pip3 install pandas && \
pip3 install openpyxl && \
pip3 install xlrd
```

Figura 2.35 Primera parte del archivo Docker.

En esta primera parte se detalla lo siguiente:

- Selección del sistema operativo base el cual será Ubuntu 18.04.
- Etiquetas del desarrollador: email del mantenedor, versión y descripción de la imagen.
- Variables de entorno: Donde se establece ‘LANG’ y ‘C.UTF-8’. Configurando la localización y el idioma en el contenedor para soportar UTF-8.
- Instalación de librerías y dependencias que serán parte del contenedor.

En la Figura 2.36, se presenta la segunda parte del archivo de Docker.

```
# Chrome
WORKDIR /tmp
RUN wget https://mirror.cs.uchicago.edu/google-chrome/pool/main/g/google-chrome-stable/google-chrome-stable_105.0.5195.102-1_amd64.deb && \
apt install -y ./google-chrome-stable_105.0.5195.102-1_amd64.deb && \
# The current version of Chrome is 105.0.5195.102
wget https://chromedriver.storage.googleapis.com/105.0.5195.52/chromedriver_linux64.zip && \
unzip chromedriver_linux64.zip && \
apt install -y poppler-utils && \
mkdir /app && \
mv chromedriver /app/chromedriver
# Go to workdir
WORKDIR /app
RUN git clone https://github.com/Zahid-Copara/microservicio_titulacion.git
RUN cp -r microservice3/* /app
RUN rm -r microservice3/
# Running microservice 3
RUN python3.6 mod1.py
CMD ["python3.6", "micro3-g.py"]
```

Figura 2.36 Segunda parte del archivo Docker

En esta parte se puede observar lo siguiente:

- Instalación de Google Chrome y ChromeDriver. En este paso es de suma importancia que se descarga la versión estable para este sistema operativo y para las versiones de las librerías instaladas.

- Configuración del directorio de trabajo y clonación del repositorio: Se va a definir el directorio de trabajo en '/app', donde se va a clonar el repositorio de microservice3.
- Ejecución del microservicio: Inicialmente, se ejecuta el script de Python 'mod1.py' durante el proceso de construcción de la imagen. Luego, se configura un comando predeterminado para que el código de 'micro-g.py' se ejecute cada vez que se inicie la imagen Docker.

Una vez definido el archivo Docker, es necesario construir su imagen. Por lo tanto, se ejecuta el comando que se observa en la Figura 2.37.

```
docker build -t microservice:latest .
```

Figura 2.37 Comando para construir una imagen Docker.

En este caso se va a construir la imagen con el nombre de 'microservice' y el tag de 'latest'. Esto presenta la salida en consola que se observa en la Figura 2.38.

Figura 2.38 Salida de la construcción de la imagen Docker.

Si el estado de la construcción de la imagen Docker es 'FINISHED' y no presenta ningún error, quiere decir que todo se ha ejecutado con normalidad y la imagen Docker se ha construido correctamente.

En este punto, se puede ejecutar a la imagen Docker con el comando de la Figura 2.39.

```
docker run -v 'local app directory':/app -it microservice:latest
```

Figura 2.39 Comando para ejecutar la imagen Docker.

En este caso, se utiliza el comando con la opción '-v' para montar un volumen local en el contenedor. Esto facilita la obtención de resultados y la modificación del código, ya que permite que el contenido de la carpeta local se refleje en la ruta '/app' del contenedor.

Para validar la salida del microservicio, es importante considerar que UPM no pudo proporcionar los enlaces a las políticas de privacidad en 27 ocasiones. Por lo tanto, para el análisis del microservicio se ha utilizado un total de 352 políticas de privacidad. El conjunto

de datos detallado, empleado en el análisis se encuentra en el ANEXO E. El rendimiento del microservicio está presente en la Tabla 2.2.

Tabla 2.2 Resultados del microservicio.

	Exitosa	Parcial	Fallida
Descarga	86,93%	9.09%	3.98%

Los resultados muestran un alto porcentaje de descargas exitosas de políticas de privacidad. Sin embargo, en los casos en los que la descarga fue parcial, se observó que muchas compañías segmentan sus políticas de privacidad en varias páginas. Esto dificulta la descarga completa, ya que la información no se encuentra en un solo lugar. En cuanto a las políticas que fallaron en la descarga, es probable que se deba a que algunas páginas web bloquean el acceso automatizado del microservicio. Se podrían realizar ajustes para evitar este problema y mejorar la disponibilidad de las políticas de privacidad.

2.3.3 Implementación del segundo módulo

En esta sección se presenta la implementación del segundo módulo, que se encarga de clasificar si una aplicación móvil recolecta datos personales.

En la Figura 2.40, se presenta el diagrama de despliegue del segundo módulo.

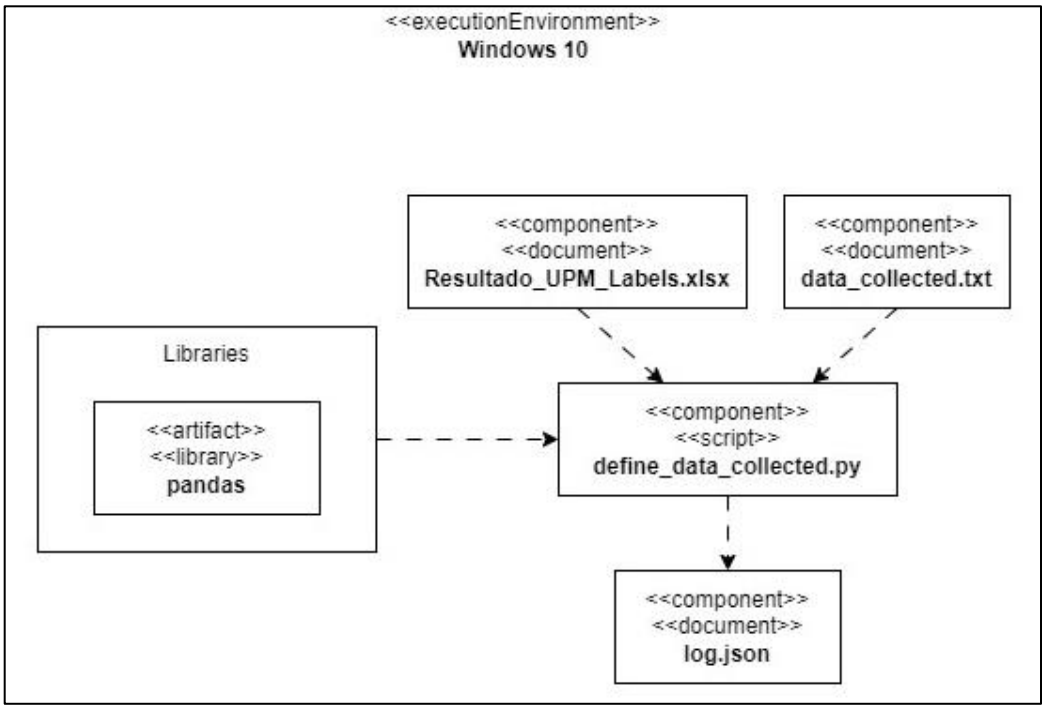

Figura 2.40 Diagrama de despliegue del segundo módulo.

Este módulo se basa en el script denominado '*define_data_collected.py*'. Para empezar, se requiere leer los archivos '*Resultado_UPM_Labels.xlsx*' y `data_collected.txt`. A continuación, En la Figura 2.41, se observa cómo se va a agrupar al conjunto de datos por el nombre del '*package_name*' y se van a enviar a la función '`check_sensitive_data()`' la cual utiliza la columna '*type_data_collected*' para comparar con la lista de datos personales, si existe coincidencia se marca a la aplicación con el valor de 'SI' para definir que recolecta datos personales y el valor de 'NO' en caso contrario.

```
10  # Crear una columna que indica si la aplicación recolecta algún dato sensible
11  def check_sensitive_data(group):
12      for data in group['type_data_collected']:
13          if data in datos_sensibles:
14              return 'SI'
15      return 'NO'
16
17  # Aplicar la función a cada grupo de aplicaciones
18  df_result = df.groupby('apk').apply(check_sensitive_data).reset_index()
```

Figura 2.41 Función que define si una aplicación recolecta o no datos personales.

Una vez obtenido el resultado se guarda la información en dos columnas, '*apk*' y '*recolect_personal_data'* y todo ello en un conjunto de datos denominado '*resultados.xlsx*'. Como se observa en la Figura 2.42.

```
19  df_result.columns = ['apk', 'recolect_personal_data']
20
21  # Guardar el resultado en un nuevo archivo Excel
22  df_result.to_excel('resultados.xlsx', index=False)
23
24  print("El archivo 'resultados.xlsx' ha sido creado con éxito.")
```

Figura 2.42 Fragmento de código que guarda la información en el conjunto de datos.

Obteniendo así los resultados ordenados de las aplicaciones móviles y si recolectan o no datos personales.

2.3.4 Implementación del tercer módulo

La implementación del tercer módulo se basa en dos scripts en Python, el primero denominado '*extract_permission.py*', el cual se basa en obtener un conjunto de datos más compacto acerca de los permisos que las aplicaciones móviles requieran. El siguiente script se denomina como '*define_permission.py*', el cual clasifica a las aplicaciones en si requieren o no de permisos sensibles a los usuarios.

A continuación, en la Figura 2.43, se presenta el diagrama del despliegue de este tercer módulo.

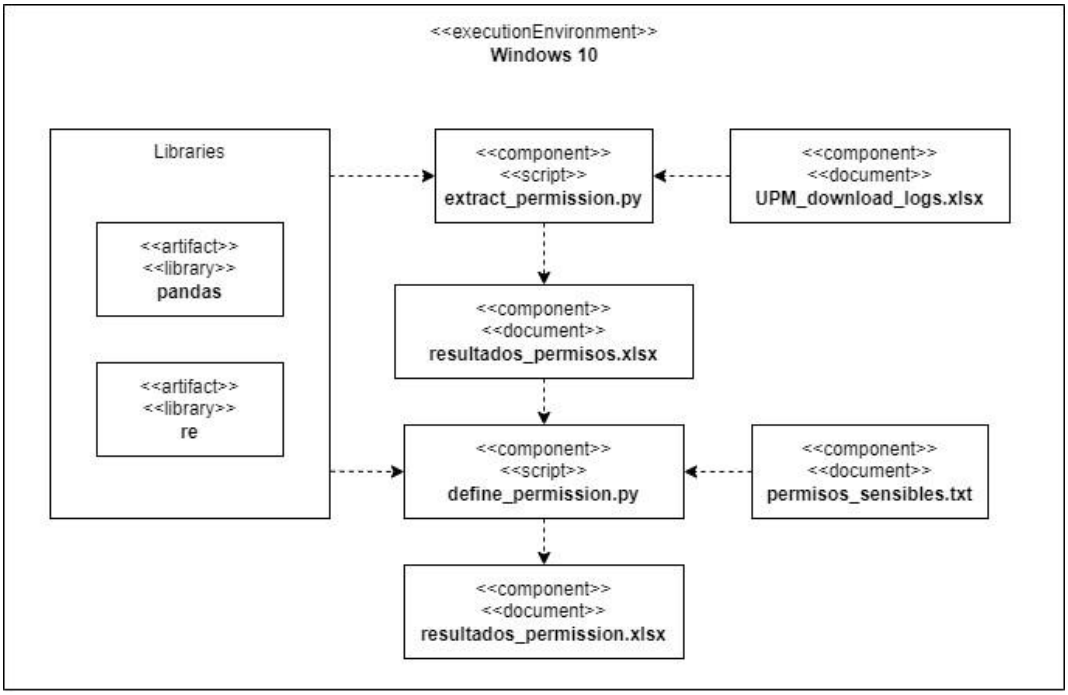

Figura 2.43 Diagrama del despliegue del tercer módulo.

El módulo inicia con la extracción de los permisos que las aplicaciones recolectan, para ello se lee el archivo que contiene esta información. Este archivo es un conjunto de datos denominado como '*UPM_Download_logs.xlsx*'. Debido a que el formato de los metadatos que contiene la información de los permisos se encuentra en un formato tipo JSON se decidió en un principio utilizar la librería de JSON para obtener estos datos. Sin embargo, debido a que el formato establecido dio problemas, se decidió mejor utilizar expresiones regulares utilizando la librería `re`.

En la Figura 2.44, se observa cómo se define la expresión regular y finalmente como se va haciendo una lista de los permisos que serán asociados a una aplicación móvil.

```
27  # Iterar sobre las filas del DataFrame
28  for index, row in df.iterrows():
29      apk = row['apk']
30      metadata = row['metadata_details']
31      if isNullorEmpty(metadata) == False:
32
33          pattern = re.compile(r'permission": \[(.*?)\]', re.DOTALL)
34          match = pattern.search(metadata.replace("'",'"'))
35
36          if match:
37              permissions_text = match.group(1)
38              permissions = [perm.strip().strip('"') for perm in permissions_text.split(',')]
39
40          # Agregar a las listas
41          for permission in permissions:
42              permission_result = permission.split('.')
43              apk_list.append(apk)
44              permissions_list.append(permission_result[-1])
```

Figura 2.44 Fragmento de código donde se aplican expresiones regulares para obtener permisos.

Una vez generada la lista de permisos, se crea un nuevo conjunto de datos donde se combina el nombre de la aplicación móvil en conjunto con el permiso obtenido. En la Figura 2.45, se puede observar cómo se genera el conjunto de datos y las columnas que se generan que en este caso son '*apk*' y '*permission*'.

```
# Crear un nuevo DataFrame con los resultados
result_df = pd.DataFrame({
    'apk': apk_list,
    'permission': permissions_list
})

# Guardar el DataFrame en un nuevo archivo Excel
result_file_path = "resultado_permisos.xlsx"
result_df.to_excel(result_file_path, index=False)
```

Figura 2.45 Fragmento de código donde se guarda el conjunto de datos resultante con los permisos.

Una vez obtenido el conjunto de datos con un formato más adecuado, se hace uso del script de Python que permite clasificar a las aplicaciones y si estas requieren de pedir permisos sensibles a los usuarios.

En la Figura 2.46, se observa la lectura que se realiza hacia dos archivos, el primero vendría a ser el conjunto de datos con el resultado de los permisos denominado '*resultado_permisos.xlsx*' y el siguiente archivo es el que contiene la lista de permisos que se consideran sensibles denominado como '*permisos_sensibles.txt*'.

```
# Leer el archivo Excel
df = pd.read_excel('resultado_permisos.xlsx')

# Leer el archivo TXT con los datos sensibles
with open('permisos_sensibles.txt', 'r') as file:
    datos_sensibles = file.read().splitlines()
```

Figura 2.46 Fragmento de código que lee los archivos de entrada.

Una vez que se han leído los archivos, se define la función '`check_permission()`', que verifica los permisos requeridos por la aplicación móvil en comparación con una lista de permisos sensibles. Si se encuentran coincidencias, la aplicación se marca con un valor de 'SI', de lo contrario, se asigna el valor 'NO'. Este proceso se ilustra en la Figura 2.47. Además, se presenta el conjunto de datos resultante, que incluye las columnas '*apk*' y *'permission'*, y se guarda con el nombre '*resultados.xlsx*'.

```python
# Crear una columna que indica si la aplicación recolecta algún dato sensible
def check_permission(group):
    for data in group['permission']:
        if data in datos_sensibles:
            return 'SI'
    return 'NO'

# Aplicar la función a cada grupo de aplicaciones
df_result = df.groupby('apk').apply(check_permission).reset_index()
df_result.columns = ['apk', 'permission']

# Guardar el resultado en un nuevo archivo Excel
df_result.to_excel('resultados.xlsx', index=False)
```

Figura 2.47 Fragmento de código de clasificación de permisos y guardado de resultados.

Al igual que en el módulo anterior, este conjunto de datos permite formar parte de los datos necesarios para el análisis de los criterios de evaluación con respecto al cumplimiento de la difusión de políticas de protección de datos por parte de las organizaciones que brindan servicios mediante las aplicaciones móviles.

2.3.5 Implementación del cuarto módulo

El cuarto módulo depende principalmente de los archivos de políticas de privacidad, esto debido a que se va a utilizar Inteligencia Artificial para analizar estás políticas y definir si cumplen o no con los requisitos que una política de privacidad debe cumplir.

Como paso previo, se compararon los archivos de políticas de privacidad proporcionados por UPM con aquellos generados mediante el microservicio. Para ello, se utilizó un script en Python que analiza ambos directorios que contienen los archivos resultantes. Se establece un tamaño mínimo de 100 bytes para cada archivo, suficiente para contener al menos una pequeña frase. Los archivos que superan este umbral se consideran como políticas descargadas exitosamente.

De los 352 enlaces de políticas de privacidad obtenidos, los porcentajes de archivos válidos fueron los siguientes:

- UPM: 91.46%
- Microservicio: 95.37%

Dado estos resultados, se opta por utilizar la salida del microservicio.

En la Figura 2.48, se presenta el diagrama de despliegue del cuarto módulo.

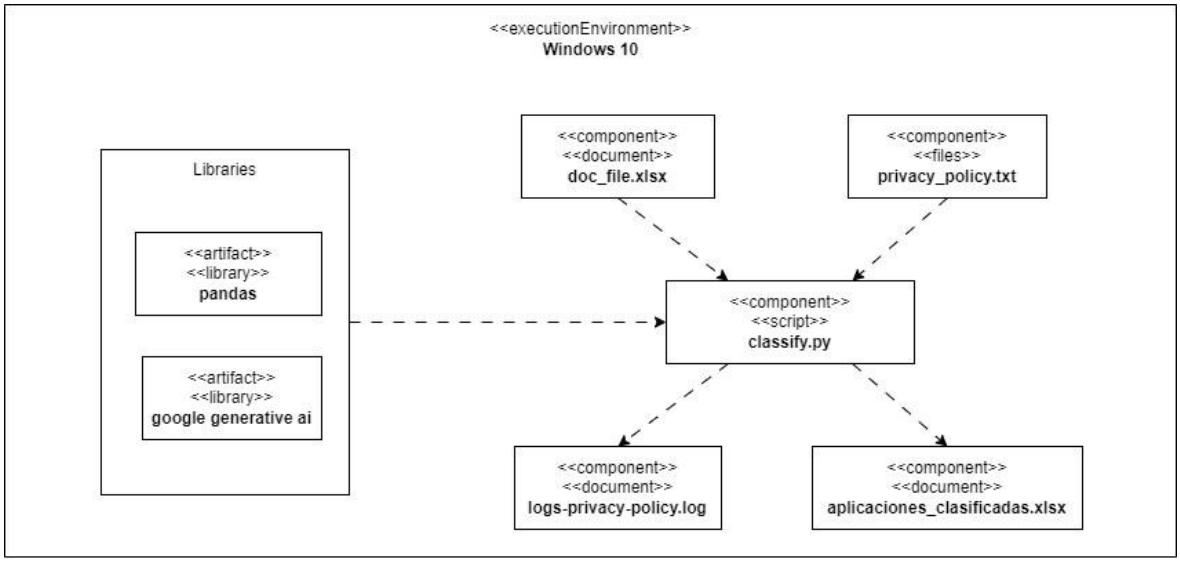

Figura 2.48 Diagrama de despliegue del cuarto módulo.

El script comienza con la lectura de dos archivos: el primero es el conjunto de datos de las aplicaciones, denominado '*doc_file.xlsx*', que contiene los nombres de las políticas de privacidad. El segundo archivo corresponde a la política de privacidad en sí, y su nombre está asociado con el nombre especificado en el conjunto de datos.

En la Figura 2.49, se visualiza el método '`process_dataset(dataset)`', donde el argumento de entrada vendría a ser el conjunto de datos que contiene la información de las aplicaciones y del nombre de los archivos de la política de privacidad. Se recorre todo el conjunto de datos para analizar aplicación por aplicación.

```
90    # Function that process dataset information
91    def process_dataset(dataset):
92        results=[]
93        count = 0
94        try:
95            for index, row in dataset.iterrows():
96                row_num = dataset.shape[0]
97                nombreApp = row['nombre_app']
98                categoria = row['categoria']
99                numeroDescargas = row['numero_descargas']
100               packageName = row['package_name']
101               ubicacion = row['ubicacion']
102               privacyPolicyUrl = row['policy_privacy_url']
103               archivoPolitica = row['policy_privacy_name']
```

Figura 2.49 Fragmento de la función que procesa el dataset.

El siguiente paso, representado en la Figura 2.50, consiste en validar el archivo de la política de privacidad. Primero, se verifica si existe un archivo asociado a la aplicación. Si no existe, se clasifica como 'NO PRESENTA' en relación con la política de privacidad. Caso contrario, se procede al flujo de trabajo y se llama a la función '*extract_data (fileName)*', donde *fileName* es el nombre del archivo de política de privacidad. Esta función lee el archivo y obtiene la información, almacenándola en una variable.

```
62  def extract_data(filename):
63      try:
64          privacyPolicyText = ''
65          current_file = files_path+'\\'+filename+'.txt'
66          if not os.path.isfile(current_file):
67              logger.error(f"file '{filename}'.txt not found")
68          with open(current_file, 'r', encoding='utf-8') as file:
69              privacyPolicyText = file.read()
70      except Exception as e:
71          reason = 'error in extract data '
72          logger.error("error in filename",
73                       extra={'exception_message':str(e), 'reason':reason})
74      else:
75          return privacyPolicyText
```

Figura 2.50 Función de extracción de la política de privacidad a partir de su archivo.

La siguiente validación implica analizar el contenido del archivo leído. Si el contenido está vacío, se clasifica la política de privacidad como 'NO PRESENTA'. Caso contrario, el flujo continúa hasta el método '*get_response(prompt)*', donde *prompt* es la instrucción enviada a la Inteligencia Artificial. La instrucción es la siguiente: "*Necesito saber si el siguiente texto entre comillas es una política de privacidad. Responde únicamente con un SI o un NO. A continuación, el texto: 'Texto de política de privacidad'*". Este método devuelve 'SI' o 'NO'. Si no se recibe respuesta, se clasifica como 'NO RESPUESTA' y se procede a su evaluación manual. Este proceso está representado en la Figura 2.51.

```
105                 privacyPolicyText = extract_data(archivoPolitica)
106                 if isNullorEmpty(privacyPolicyText) == False:
107                     prompt = f"Necesito saber si el siguiente texto que se encuentra entre comillas es una política de privacidad. Responde únicamente con un SI o un NO.
108                     policyPrivacyClassification = get_response(prompt)
109                     if policyPrivacyClassification:
110                         results.append((nombreApp, categoria, numeroDescargas, packageName, ubicacion, archivoPolitica, privacyPolicyUrl, policyPrivacyClassification))
111                     else:
112                         logger.error("error while receiving gemini info")
113                         print("No response from gemini")
114                         results.append((nombreApp, categoria, numeroDescargas, packageName, ubicacion, archivoPolitica, privacyPolicyUrl, 'NO RESPUESTA'))
115                     time.sleep(5)
116                 else:
117                     results.append((nombreApp, categoria, numeroDescargas, packageName, ubicacion, archivoPolitica, privacyPolicyUrl, 'NO PRESENTA'))
118             else:
119                 results.append((nombreApp, categoria, numeroDescargas, packageName, ubicacion, archivoPolitica, privacyPolicyUrl, 'NO PRESENTA'))
120             count = count + 1
121             print(f"{count}/{row_num} processed!")
122     except Exception as e:
123         reason = 'error while proccesing dataset'
124         logger.error("error in processing dataset",
125                      extra={'exception_message':str(e), 'reason':reason})
126         results.append((nombreApp, categoria, numeroDescargas, packageName, ubicacion, archivoPolitica, privacyPolicyUrl, 'NO RESPUESTA'))
127     else:
128         return results
```

Figura 2.51 Fragmento de código que presenta la validación y llamada a la Inteligencia Artificial.

Finalmente, en la Figura 2.52, se presenta la función '*save_dataset(results)*', que toma como argumento los resultados generados por el método anterior, incluyendo las

etiquetas de las políticas de privacidad. Esta función guarda los datos en las mismas columnas de entrada, añadiendo una columna adicional denominada '*is_policy_ privacy*'. Esta columna incluye los valores posibles: 'NO', 'SI', 'NO PRESENTA' y 'NO RESPUESTA'. El conjunto de datos actualizado se guarda con el nombre *aplicaciones_clasificadas.xlsx*.

```
129 def save_dataset(results):
130     results_df = pd.DataFrame(results, columns=['nombre_app','categoria','numero_descargas','package_name',
131     output_file = actual_directory+'\\aplicaciones_clasificadas.xlsx'
132     results_df.to_excel(output_file, index=False)
```

Figura 2.52 Función que permite guardar el nuevo conjunto de datos.

Dado que se utilizó la inteligencia artificial de Google Gemini para identificar políticas de privacidad, es necesario validar la precisión de esta IA para determinar si en realidad ha identificado correctamente las políticas de privacidad. Por ello, se ha decidido usar una muestra de 189 políticas de privacidad para realizar un análisis manual. Esta cantidad se seleccionó debido a que representa más de la mitad de las políticas de privacidad extraídas. Los resultados de este análisis se compararon con los proporcionados por Gemini para evaluar la precisión de la IA en la identificación de políticas de privacidad.

Originalmente Gemini dio los siguientes resultados con la muestra seleccionada:

- 166 son políticas de privacidad
- 23 no son políticas de privacidad

Con un análisis manual de las políticas se encuentran los siguientes resultados:

- 151 son políticas de privacidad
- 38 no son políticas de privacidad

La nomenclatura para el cálculo de las métricas son las siguientes:

- Verdadero Positivo (TP)
- Falso Positivo (FP)
- Verdadero Negativo (TN)
- Falso Negativo (FN)

El cálculo de métricas se realiza de la siguiente forma:

- TP = 151
- FP = 166 – 151 = 15
- TN = 38
- FN = 38 – 23 = 15

Los verdaderos positivos son los 151 casos correctamente identificados como políticas de privacidad. Los falsos positivos se calculan como la diferencia entre los 166 casos que Gemini clasificó como políticas de privacidad y los 151 verdaderos positivos, resultando en 15 falsos positivos. Los verdaderos negativos corresponden a los 38 casos que se clasificaron manualmente como no políticas de privacidad. Los falsos negativos se obtienen restando estos 38 verdaderos negativos de los casos que la IA clasificó incorrectamente como políticas de privacidad, resultando en 15 falsos negativos.

A continuación, se realiza el cálculo de:

$$Precisión = \frac{TP}{TP + FP} = \frac{151}{151 + 15} = 90.96\% \quad (2.1)$$

$$Exactitud = \frac{TP + TN}{TP + FP + TN + FN} = \frac{151 + 38}{151 + 15 + 15 + 38} = 86.3\% \quad (2.2)$$

$$Sensibilidad = \frac{TP}{TP + FN} = \frac{151}{151 + 15} = 90.96\% \quad (2.3)$$

$$Especificidad = \frac{TN}{TN + FP} = \frac{38}{38 + 15} = 71.69\% \quad (2.4)$$

Una vez realizado el cálculo de las métricas, se requiere de una interpretación:

- Precisión: La IA tiene una alta precisión (90.96%) al clasificar documentos como políticas de privacidad, lo que significa que la mayoría de los documentos que identificó como tales son efectivamente políticas de privacidad.
- Exactitud: La IA tiene una exactitud considerablemente alta igualmente (86.3%) al clasificar tanto documentos de políticas de privacidad como aquellos que no lo son.
- Sensibilidad: La IA identifica correctamente el 90.96% de los documentos que son realmente políticas de privacidad.
- Especificidad: El valor de especificidad es del 71.69%, lo que indica que el modelo identifica correctamente el 71.69% de las políticas de privacidad que no cumplen con los requisitos. Esto significa que el modelo tiene una capacidad razonable para detectar correctamente las políticas que no cumplen, pero también tiene un 28.31% de falsos positivos.

2.3.6 Implementación del quinto módulo

La implementación del quinto módulo se basa principalmente en el poder clasificar si existe un potencial cumplimiento de la difusión de políticas de protección de datos por parte de las organizaciones que brindan servicios mediante aplicaciones móviles según la LOPDP.

Una vez obtenida toda la información necesaria para formar un conjunto de datos sólido, se puede definir condiciones que permitirán obtener los resultados.

En la Figura 2.53 se presenta el diagrama del despliegue de este quinto módulo.

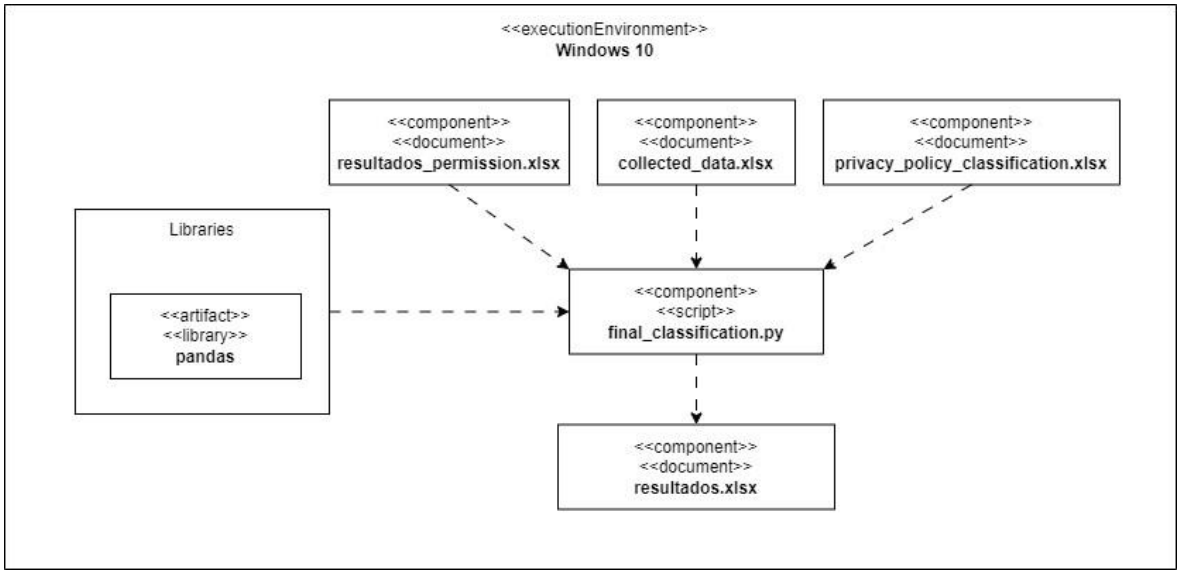

Figura 2.53 Diagrama de despliegue del quinto módulo.

Se procede a la lectura de los tres conjuntos de datos generados por los módulos anteriores. El primer conjunto de datos, denominado '*privacy_policy_classification.xlsx'*, proviene del cuarto módulo, que contiene la información sobre las políticas de privacidad. El segundo conjunto de datos, '*collected_data.xlsx'*, se refiere al segundo módulo, donde se determinó si una aplicación recolecta datos personales. Finalmente, el tercer conjunto de datos, '*resultados_permission.xlsx'*, corresponde al tercer módulo, que establece si una aplicación requiere permisos sensibles de los usuarios. Este proceso se ilustra en la Figura 2.54.

```
3   # Leer los datasets desde los archivos Excel
4   privacy_policy_classification = pd.read_excel('privacy_policy_classification.xlsx')
5   collected_data = pd.read_excel('collected_data.xlsx')
6   permission_data = pd.read_excel('resultados_permission.xlsx')
```

Figura 2.54 Lectura de los conjuntos de datos necesarios para el módulo.

En la Figura 2.55, se presenta el método `rename` para poder cambiar el nombre de las columnas para los conjuntos de datos de los permisos sensibles y datos personales. Esto para que tengan el mismo nombre con respecto al archivo de las políticas de privacidad y se puedan relacionar los tres archivos de manera sencilla.

```
# Renombrar la columna 'apk' a 'package_name' en el dataset 'collected_data' para poder hacer el merge
collected_data = collected_data.rename(columns={'apk': 'package_name'})
permission_data = permission_data.rename(columns={'apk': 'package_name'})
```

Figura 2.55 Fragmento de código que permite renombrar las columnas.

El siguiente paso consiste en procesar los conjuntos de datos. Primero, se combinarán los tres conjuntos de datos utilizando la columna '*package_name'* como clave, para consolidar toda la información en un solo conjunto de datos. Para lograr esto, se emplea la función '`merge()`' de la librería pandas. Posteriormente, se rellenan los valores faltantes en las columnas '*recolect_personal_data*' y '*permission*' con el valor 'NO'. Esto se puede observar en la Figura 2.56.

```
# Realizar el merge de los datasets basado en 'package_name'
merged_data_collected = pd.merge(privacy_policy_classification, collected_data, on='package_name', how='left')

# Rellenar los valores faltantes en 'recolect_personal_data' con 'NO'
merged_data_collected['recolect_personal_data'] = merged_data_collected['recolect_personal_data'].fillna('NO')

# Realizar el merge del dataset resultante basado en 'package_name'
merged_data = pd.merge(merged_data_collected, permission_data, on='package_name', how='left')

# Rellenar los valores faltantes en 'permission' con 'NO'
merged_data['permission'] = merged_data['permission'].fillna('NO')
```

Figura 2.56 Fragmento de código que da tratamiento al conjunto de datos.

A continuación, la Figura 2.57 muestra los criterios utilizados para el definir el potencial cumplimiento de la difusión de políticas de protección de datos por parte de las organizaciones que brindan servicios mediante aplicaciones móviles en Ecuador. Estos criterios están detallados en la Tabla 2.1 y se definen dentro de la función '`cumple_lopd(row)`', en la cual el parámetro de entrada es cada fila para analizar.

```
def cumple_lopd(row):
    if row['recolect_personal_data'] == 'NO' and row['permission'] == 'NO':
        if row['is_policy_privacy'] == 'SI':
            return 'NO APLICA'
        elif row['is_policy_privacy'] == 'NO' or row['is_policy_privacy'] == 'NO PRESENTA':
            return 'NO APLICA'

    if row['recolect_personal_data'] == 'NO' and row['permission'] == 'SI':
        if row['is_policy_privacy'] == 'SI':
            return 'NO'
        elif row['is_policy_privacy'] == 'NO':
            return 'NO'
        elif row['is_policy_privacy'] == 'NO PRESENTA':
            return 'NO'

    if row['recolect_personal_data'] == 'SI' and row['permission'] == 'NO':
        if row['is_policy_privacy'] == 'SI':
            return 'SI'
        elif row['is_policy_privacy'] == 'NO' or row['is_policy_privacy'] == 'NO PRESENTA':
            return 'NO'

    if row['recolect_personal_data'] == 'SI' and row['permission'] == 'SI':
        if row['is_policy_privacy'] == 'SI':
            return 'SI'
        elif row['is_policy_privacy'] == 'NO' or row['is_policy_privacy'] == 'NO PRESENTA':
            return 'NO'

    return 'NO APLICA'

merged_data['cumple_lopd'] = merged_data.apply(cumple_lopd, axis=1)
```

Figura 2.57 Fragmento de código con las condiciones para clasificar el potencial cumplimiento de la LOPDP.

Finalmente, se guardan los resultados dentro de un nuevo conjunto de datos denominado '*resultados.xlsx*'.

2.3.7 Criterios para la selección de aplicaciones

En esta sección se buscan aquellos criterios para la selección de un conjunto de aplicaciones móviles que serán evaluadas usando el método y módulos antes descritos. A continuación, se describen brevemente estos criterios:

i. **Variedad de sensores y datos:** Los dispositivos móviles contienen una amplia variedad de sensores que pueden considerarse sensibles en cuanto a la privacidad del usuario. Entre estos se incluyen sensores de GPS, cámaras, micrófonos y aquellos relacionados con la salud, como los sensores de BPM y presión sanguínea [31]. Además, se encuentran estructuras de datos como las listas de contactos o el almacenamiento local de archivos del smartphone. Esta abundancia de información puede representar una vulneración de la privacidad de los usuarios, razón por la cual se dará prioridad a aquellas categorías de aplicaciones que soliciten permisos varios para acceder a sensores y estructura de datos sensibles.
ii. **Categorías:** Este criterio es importante porque permite discernir qué aplicaciones son más propensas a manejar información delicada o que podría comprometer la privacidad del usuario. Por ejemplo, una aplicación de linterna y una aplicación de salud difieren significativamente en cuanto a la sensibilidad de los datos que manejan. Mientras que la primera es poco probable que procese datos personales, la segunda puede incluir información médica sensible.
iii. **Popularidad:** La popularidad de una aplicación se puede determinar identificando cuáles han sido las más descargadas. Este criterio es relevante porque a más popularidad mayor riesgo de afectación a la privacidad, pues una aplicación ampliamente descargada y que incumpla un requerimiento afectaría a un mayor número de usuarios.
iv. **País de origen:** Este criterio permite incluir tanto aplicaciones ecuatorianas como aplicaciones extranjeras, cuyo público objetivo sean los ciudadanos ecuatorianos. Este criterio es relevante porque permitirá. Este criterio es relevante porque permitirá identificar diferencias en las prácticas de privacidad según la región. Además, la inclusión de aplicaciones extranjeras ayudará a evaluar cómo se gestionan las políticas de privacidad y el cumplimiento de las normativas ecuatorianas.

2.3.8 Selección de las aplicaciones móviles

Basados en los cuatro criterios definidos en la subsección anterior, a continuación, se describe el procedimiento para la selección de las aplicaciones móviles.

Para obtener un panorama de las aplicaciones móviles de Ecuador, se utilizó la aplicación Android 'Las apps de Ecuador', que, hasta el 19 de marzo de 2024, registraba un total de 730 aplicaciones móviles. Se aplicaron dos filtros para afinar la selección: primero, se consideró la popularidad local. Dado que Ecuador tiene una población relativamente pequeña y podría haber aplicaciones menos conocidas internacionalmente que sean populares en el país, se identificaron 137 aplicaciones ecuatorianas con este criterio. Segundo, se enfocó en la categoría 'Financiera' de Google Play, encontrando 12 aplicaciones de empresas aseguradoras que podrían no haber sido incluidas en el primer grupo. Estas aplicaciones se obtuvieron de la página de la Corporación del Seguro de Depósitos (COSEDE) [33]. En total, se han identificado 149 aplicaciones ecuatorianas de un total de 145 empresas ecuatorianas.

Para las aplicaciones extranjeras, se aplicó el criterio de popularidad en las siguientes categorías que más probabilidad tengan de procesar o tratar datos personales: Salud, Finanzas, Social, Entretenimiento y Videojuegos, resultando en una selección de 132 aplicaciones. Además, basándose en la información proporcionada por el Mercado de Valores [34], se identificaron 98 aplicaciones móviles de empresas extranjeras con operaciones en Ecuador. En total, se han registrado 230 aplicaciones extranjeras.

En total, se dispone de 379 aplicaciones para su análisis las cuales se encuentran en el ANEXO H. Las aplicaciones móviles se encuentran dentro de un archivo Excel, donde se tiene la siguiente información: Nombre de la aplicación, Categoría, Número de descargas, Nombre de paquete y Ubicación.

2.3.9 Obtener los datos y metadatos de las aplicaciones

La obtención de los datos de las aplicaciones antes descritas se realizó mediante los módulos descritos en la sección anterior. Los datos descargados están disponibles en el ANEXO F para las etiquetas de privacidad y en el ANEXO G para los metadatos de las aplicaciones.

3 RESULTADOS Y DISCUSIÓN

En este capítulo, se presentan los resultados obtenidos por el último módulo, debido a que este es el que contiene los resultados con respecto al potencial cumplimiento o incumplimiento de la difusión de políticas de protección de datos por parte de las organizaciones que brindan servicios mediante aplicaciones móviles en Ecuador. En el ANEXO H se puede visualizar los datos resultantes.

En primer lugar, se analiza la relación entre el potencial cumplimiento de las políticas de privacidad por parte de las organizaciones ecuatorianas y extranjeras. Este análisis proporciona una visión del estado actual del cumplimiento en Ecuador con relación con la LOPDP.

En la Figura 3.1, se observa la clasificación del potencial cumplimiento de las empresas ecuatorianas con respecto a la difusión de políticas de privacidad.

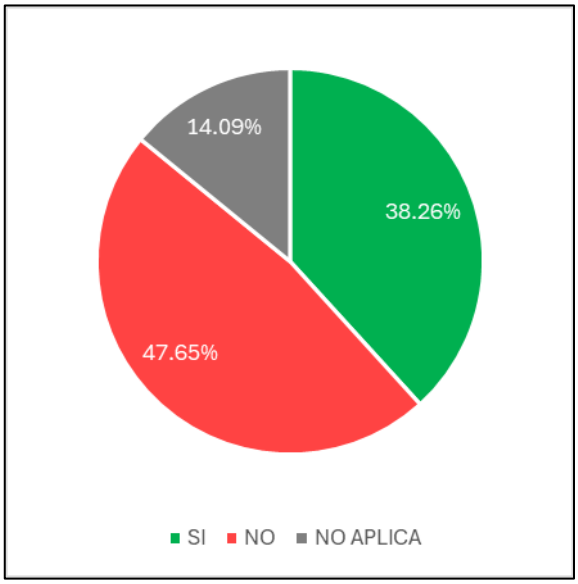

Figura 3.1 Tasa del potencial cumplimiento de las empresas ecuatorianas.

En el ámbito ecuatoriano, al analizar un total de 145 empresas, se observa un potencial de incumplimiento, alcanzando un 47.65%. En contraste, solo el 38.26% muestra un potencial cumplimiento. Además, se encuentra una etiqueta de 'NO APLICA' con el 14.09%, que indica que la aplicación móvil no realizaría tratamiento de datos personales (basado en la declaración realizada en la etiqueta de privacidad y los permisos solicitados), por lo que una política de privacidad no sería necesaria.

En la Figura 3.2, se muestra la distribución del potencial cumplimiento entre las empresas extranjeras.

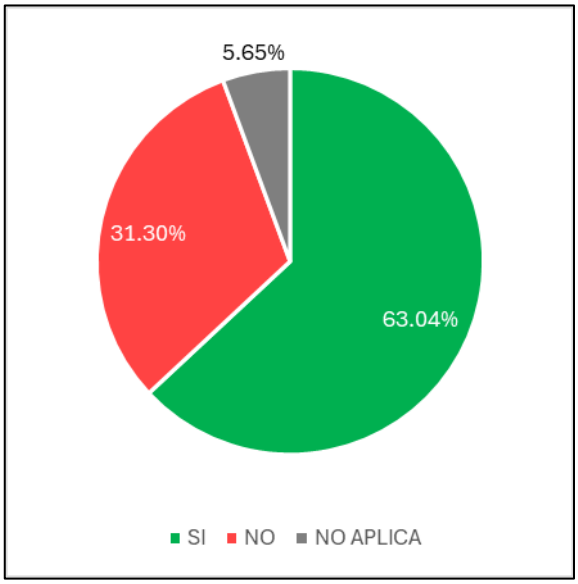

Figura 3.2 Tasa del potencial cumplimiento de las empresas extranjeras.

En el caso de las organizaciones extranjeras, se observa que la mayoría presenta un potencial cumplimiento en la difusión de políticas de privacidad, con un 63.04%. En contraste, el 31.30% muestra un potencial incumplimiento, mientras que el 5.65% está etiquetado como 'NO APLICA'.

En resumen, con respecto a la ubicación geográfica se revelan diferencias significativas en el cumplimiento de las políticas de privacidad. Mientras que las organizaciones extranjeras en general demuestran un mejor cumplimiento, Ecuador enfrenta desafíos notables en esta área. Esto subraya la necesidad urgente de fortalecer el cumplimiento de los requisitos relacionados con la difusión de políticas de privacidad en el país.

La siguiente área de interés se centra en las categorías de aplicaciones móviles en Google Play. Este análisis permite evaluar el potencial cumplimiento de las políticas de privacidad en aquellas categorías con mayor probabilidad de tratar datos personales. Al igual que en el punto anterior, se examinarán las categorías y su potencial cumplimiento tanto en aplicaciones de organizaciones ecuatorianas como extranjeras.

En la Figura 3.3, se observa la gráfica que relaciona las categorías y el potencial cumplimiento en las empresas ecuatorianas.

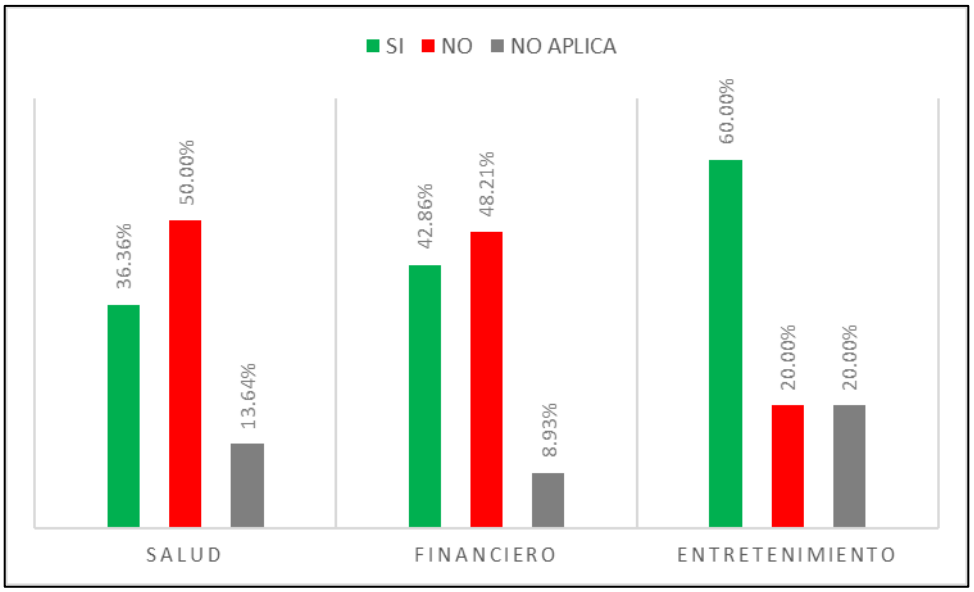

Figura 3.3 Tasa del potencial cumplimiento de las organizaciones ecuatorianas, según la categoría de sus apps.

En relación con las categorías más propensas a tratar datos personales en Ecuador, se observa que la mayoría presenta un potencial incumplimiento. La categoría con el menor nivel de cumplimiento es 'Salud', con solo un 36.36%. Le sigue la categoría 'Financiero', con un 42.86% de cumplimiento. Por otro lado, la categoría 'Entretenimiento' muestra el mayor porcentaje de potencial cumplimiento, alcanzando el 60%.

En la Figura 3.4 se presenta una gráfica que muestra el potencial cumplimiento de las políticas de privacidad en relación con las categorías de Google Play, específicamente para las aplicaciones móviles de origen extranjero.

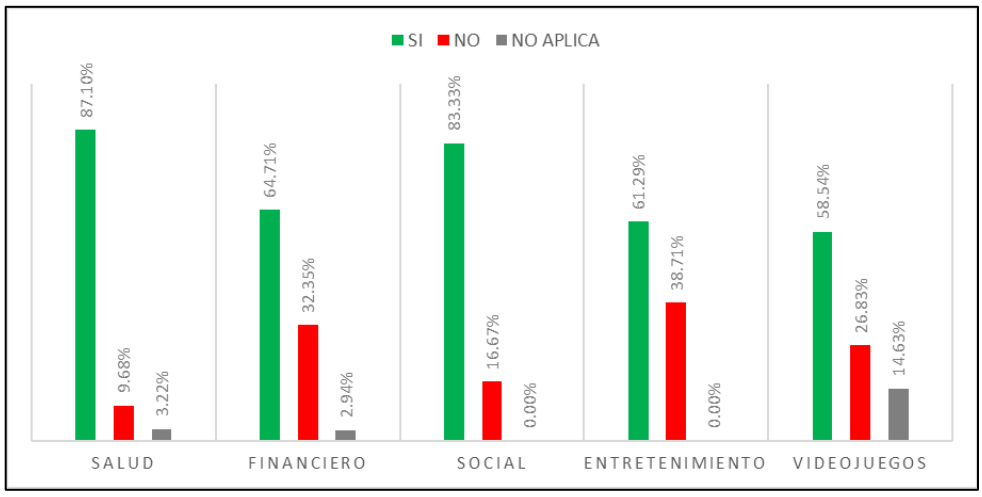

Figura 3.4 Tasa del potencial cumplimiento de las organizaciones extranjeras, según la categoría de sus apps.

La principal diferencia entre el cumplimiento de políticas de privacidad en las organizaciones ecuatorianas y extranjeras radica en que, en Ecuador, la mayoría de las categorías muestran un potencial incumplimiento. En contraste, las organizaciones extranjeras presentan un alto porcentaje de potencial cumplimiento en casi todas las categorías. Las categorías extranjeras presentan los siguientes niveles de cumplimiento: 'Salud' con un 87.10%, 'Financiero' con un 64.71%, 'Social' con un 83.33%, 'Entretenimiento' con un 61.29%, y 'Videojuegos' con un 58.54%. En general, estas categorías superan el nivel de cumplimiento potencial observado en las categorías dentro de Ecuador. Además, cabe destacar que las categorías 'Social' y 'Videojuegos' no se analizaron en el ámbito ecuatoriano debido a la limitada cantidad de aplicaciones disponibles en estas categorías en Ecuador.

Otra etiqueta de interés es el número de descargas, ya que permite correlacionar la popularidad de las aplicaciones móviles con el potencial cumplimiento de las políticas de protección de datos por parte de las organizaciones que las ofrecen. En este análisis, se lleva a cabo una evaluación general comparativa entre las organizaciones ecuatorianas y extranjeras.

En la Figura 3.5 se muestra un gráfico de barras que ilustra la relación entre el número de descargas y el potencial cumplimiento de las políticas de privacidad. Este gráfico se centra exclusivamente en el potencial cumplimiento y no incluye datos sobre incumplimiento o la etiqueta 'NO APLICA'.

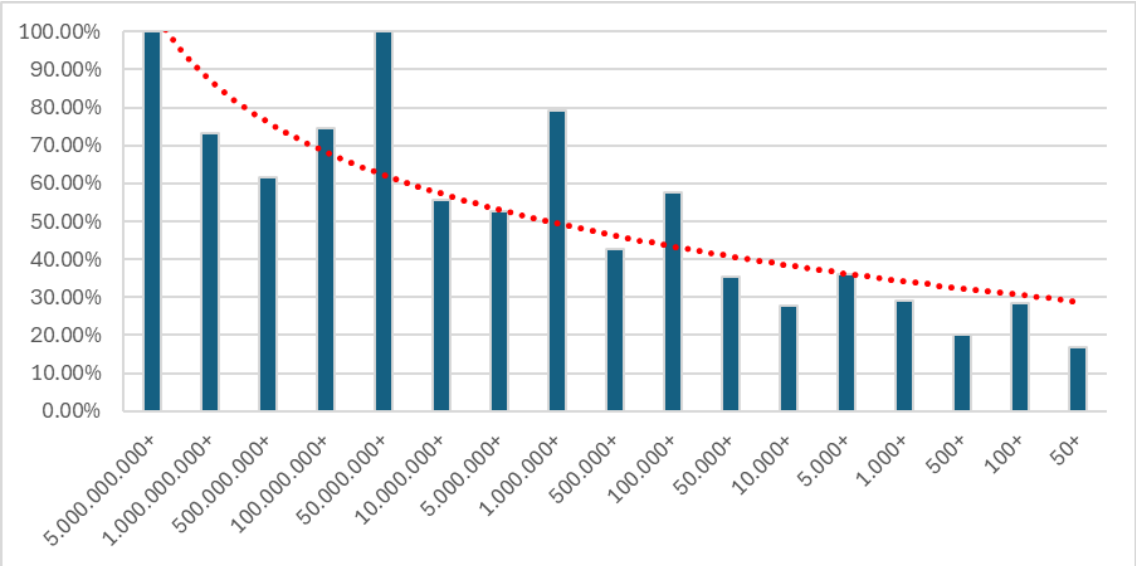

Figura 3.5 Gráfica de barras que relaciona la popularidad de las aplicaciones y el potencial cumplimiento de las organizaciones.

Se observa una correlación entre el número de descargas (popularidad) y el potencial cumplimiento de las políticas de privacidad. Las aplicaciones más populares, con más de 5 mil millones de descargas, alcanzan un potencial cumplimiento del 100%, mientras que las menos populares, con apenas más de 50 descargas, tienen un potencial cumplimiento cercano al 20% Esto permite considerar que las aplicaciones más populares son más probables de tener un mayor potencial de cumplimiento por parte de las organizaciones, mientras que las aplicaciones menos conocidas parecen tener un menor cumplimiento. Por lo tanto, se identifica una tendencia descendente en la relación entre la popularidad de las aplicaciones y su potencial cumplimiento en la difusión de políticas de privacidad. Esto podría deberse a que las aplicaciones más populares provienen de organizaciones con mayores recursos humanos y económicos, que facilitan la creación de equipos multidisciplinares (legal y técnico) que se aseguran de garantizar el cumplimiento con normativas de protección de datos. Además, las aplicaciones más populares suelen ser de origen extranjero, donde las normativas de protección de datos han estado vigentes durante varios años, a diferencia de Ecuador, que es relativamente reciente.

En cuanto a los datos personales, es posible identificar cuáles son los tipos de datos más recolectados por las aplicaciones móviles. La Figura 3.6 muestra una gráfica que ilustra los datos personales más frecuentemente recopilados por estas aplicaciones.

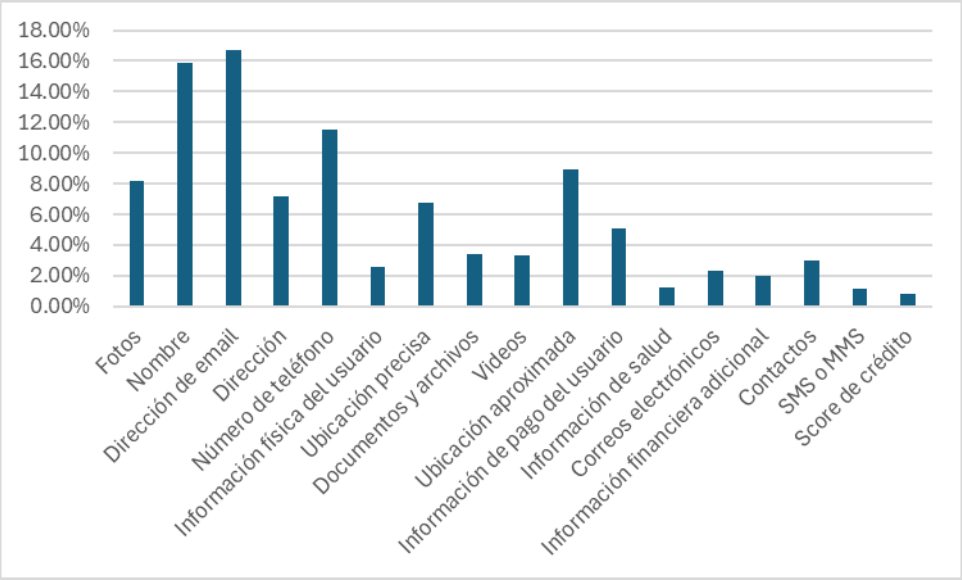

Figura 3.6 Gráfico de barras con los datos personales que más recolectan las aplicaciones móviles.

Se observa que los cinco datos personales más recolectados por las aplicaciones móviles son: dirección de email, nombre, número de teléfono, ubicación aproximada y fotos. La

Tabla 3.1 presenta un desglose de todos los datos personales recolectados, junto con el porcentaje que indica la frecuencia con la que se recopilan.

Tabla 3.1 Datos personales recolectados y su frecuencia de recopilación.

Dirección de email	16.74%
Nombre	15.86%
Número de teléfono	11.54%
Ubicación aproximada	8.90%
Fotos	8.19%
Dirección	7.14%
Ubicación precisa	6.78%
Información de pago del usuario	5.11%
Documentos y archivos	3.44%
Videos	3.35%
Contactos	3.00%
Información física del usuario	2.56%
Correos electrónicos	2.29%
Información financiera adicional	1.94%
Información de salud	1.23%
SMS o MMS	1.15%
Score de crédito	0.79%

4 CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones y recomendaciones que se han obtenido durante el desarrollo del presente trabajo.

4.1 Conclusiones

- Se ha observado que Ecuador está rezagado en comparación con el ámbito extranjero en cuanto al potencial cumplimiento en la difusión de políticas de privacidad. Esta diferencia puede atribuirse a que, en el extranjero, las políticas de protección de datos han estado en vigor durante varios años, lo que ha permitido un mayor desarrollo y cumplimiento en comparación con Ecuador.
- La revisión de los fundamentos teóricos basados en todo lo relacionado a la protección de datos personales fue de gran ayuda. Principalmente para entender cuando las empresas dueñas de las aplicaciones móviles realmente podrían estar cumpliendo o incumpliendo con la difusión de políticas de protección de datos.

- Los criterios de selección para las aplicaciones móviles como popularidad, categoría y ubicación geográfica permitieron identificar una muestra representativa y sólida tanto a nivel ecuatoriano como internacional.
- La selección de criterios de evaluación para determinar si las empresas propietarias de aplicaciones móviles cumplen con la difusión de políticas de protección de datos, según la Ley Orgánica de Protección de Datos Personales, es un paso crucial. Este proceso facilita el análisis de las políticas de privacidad, asegurando que cumplan con los criterios de evaluación de manera más eficiente y efectiva.
- Los artefactos de software fueron capaces de obtener de manera satisfactoria toda la información necesaria para analizar los criterios de evaluación que posteriormente permitían el definir el potencial cumplimiento por parte de las empresas dueñas de las aplicaciones. Esta recopilación fue posible gracias a las adaptaciones realizadas en el microservicio del primer módulo y en el resto de los módulos, que facilitaron el manejo de diferentes conjuntos de datos y optimizaron el análisis de la información.
- Se ha podido encontrar una relación entre la popularidad de las aplicaciones móviles y el potencial cumplimiento en la difusión de políticas de privacidad. Las aplicaciones más populares muestran un mayor cumplimiento en comparación con las menos populares. Esto se puede deber a que las aplicaciones más populares pueden tener mayores recursos y mayor enfoque en la transparencia para mantener la confianza de sus usuarios. En contraste, las aplicaciones menos populares, con menos visibilidad y recursos, pueden no priorizar tanto la implementación y difusión efectiva de sus políticas de privacidad.
- Aunque la Ley Orgánica de Protección de Datos Personales está en vigor desde mayo de 2021, resulta preocupante que muchas aplicaciones móviles en el contexto ecuatoriano aún no cumplan con la difusión de políticas de privacidad. Solo un 38.26% de las empresas ecuatorianas parecen cumplir con estos requisitos, mientras que, aunque el 63.04% de las empresas extranjeras muestra una mejora, este porcentaje sigue siendo bajo. Esta situación refleja una posible falta de interés en proteger la privacidad de los usuarios y sus datos personales.

4.2 Recomendaciones

- En el cuarto módulo del servicio de Gemini, se recomienda generar múltiples claves de proyecto para maximizar el uso de la capa gratuita de la Inteligencia Artificial. Esto permitirá procesar todas las instrucciones necesarias para obtener resultados completos en la clasificación de las políticas de privacidad.
- Se recomienda utilizar conjuntos de datos para procesar los resultados en los módulos. Esta práctica facilita la identificación del potencial cumplimiento con la LOPDP, especialmente al usar scripts en Python junto con la biblioteca pandas, lo que optimiza el análisis y mejora la eficiencia del proceso

5 REFERENCIAS BIBLIOGRÁFICAS

[1] A. Tecnología/EP. “Los móviles Android recopilan 20 veces más datos que los iPhone, según un estudio”. Diario ABC. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.abc.es/tecnologia/moviles/telefonia/abci-moviles-android-recopilan-20-veces-mas-datos-iphone-segun-estudio-202104051107_noticia.html

[2] INEC. “Tecnologías de la Información y Comunicación-TIC”. Instituto Nacional de Estadística y Censos. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.ecuadorencifras.gob.ec/tecnologias-de-la-informacion-y-comunicacion-tic/

[3] D. S. Guamán, D. Rodriguez, J. M. del Alamo y J. Such, “Automated GDPR compliance assessment for cross-border personal data transfers in android applications”, *ScienceDirect*, vol. 130, 2023. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://doi.org/10.1016/j.cose.2023.103262

[4] PRESIDENCIA DE LA REPÚBLICA DEL ECUADOR. “Ley orgánica de protección de datos”. Corporación Nacional de Finanzas Populares y Solidarias – CONAFIPS – Impulsamos la Economía Popular y Solidaria en todo el país. Accedido el 6 de agosto de 2024. [En línea].
Disponible: https://www.finanzaspopulares.gob.ec/wp-content/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

[5] L. Ponce. “Todo lo que debes conocer sobre la Protección de Datos Personales”. PwC. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.pwc.ec/es/entrevistas-de-temas-de-interes/todo-lo-que-debes-conocer-sobre-la-proteccion-de-datos-personales.html

[6] H. F. Costa. “Ley de Protección de Datos Personales, un nuevo estándar de seguridad y privacidad para el Ecuador”. Forbes Ecuador: Nada personal, solo negocios - Forbes Ecuador. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.forbes.com.ec/columnistas/ley-proteccion-datos-personales-nuevo-estandar-seguridad-privacidad-ecuador-n8975

[7] M. Arboleda. “Estadísticas digitales Ecuador 2021 por Datareportal - BrandEc”. BrandEc - Blog de Mauricio Arboleda Acosta. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://hablemosdemarcas.com/estadisticas-digitales-ecuador-2021/

[8] CEPAL. “Gestión de datos de investigación”. Biblioguías - Biblioteca de la CEPAL. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://biblioguias.cepal.org/c.php?g=495473&p=4398114

[9] Cloudflare. “¿Qué es la privacidad de los datos?” Cloudflare. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://www.cloudflare.com/es-es/learning/privacy/what-is-data-privacy/

[10] J. M. del Alamo, D. S. Guamán, B. García y A. Diez, “A systematic mapping study on automated analysis of privacy policies”, *Spring Link.*, vol. 104, 2022. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://doi.org/10.1007/s00607-022-01076-3

[11] D. Rodriguez, I. Yang, J. M. del Alamo y N. Sadeh, “Large language models: A new approach for privacy policy analysis at scale”, *ArXiv*, 2024. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://doi.org/10.48550/arXiv.2405.20900

[12] Android. “Qué es android?” Android. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.android.com/intl/es_es/what-is-android/

[13] M. F. Melo. “Android vs. iOS: El mapa mundial de Android e iOS”. statista. Accedido el 5 de agosto de 2024. [En línea].
Disponible: https://es.statista.com/grafico/29620/sistema-operativo-movil-con-la-mayor-cuota-de-mercado-por-pais/

[14] “Number of android apps on google play”. AppBrain. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://www.appbrain.com/stats/number-of-android-apps

[15] J. C. Miranda Carrasco, “PII-DETRI-2021-06: Mejora de las técnicas de evaluación de la integridad contextual de la privacidad de las aplicaciones móviles: desarrollo de microservicios para la extracción de políticas de privacidad desde artefactos de una aplicación móvil Android.”, Bachelor Thesis, EPN, Quito, 2022. Accedido el 5 de agosto de 2024. [En línea].
Disponible: https://bibdigital.epn.edu.ec/bitstream/15000/23248/1/CD%2012661.pdf

[16] “Permisos en Android | Android Developers”. Android Developers. Accedido el 6 de agosto de 2024. [En línea].
Disponible: https://developer.android.com/guide/topics/permissions/overview?hl=es-419

[17] “Manifest.permission | Android Developers”. Android Developers. Accedido el 6 de agosto de 2024. [En línea].
Disponible: https://developer.android.com/reference/android/Manifest.permission

[18] M. A. Revelo Báez, “Evaluación de prácticas de privacidad en aplicaciones móviles: desarrollo de un módulo de etiquetado de prácticas de recolección de datos personales en políticas de privacidad en español usando técnicas PLN y aprendizaje automático.”, Bachelor Thesis, EPN, Quito, 2023. Accedido el 5 de agosto de 2024. [En línea].
Disponible: http://bibdigital.epn.edu.ec/handle/15000/24757

[19] J. M. del Alamo, D. S. Guamán, B. Balmori y A. Diez, “Privacy Assessment in Android Apps: A Systematic Mapping Study”, *Electronics*, n. º 16, 2021. Accedido el 5 de agosto de 2024. [En línea].
Disponible: https://doi.org/10.3390/electronics10161999

[20] “¿Qué es Python? - Explicación del lenguaje Python - AWS”. Amazon Web Services, Inc. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://aws.amazon.com/es/what-is/python/

[21] “Pandas: La biblioteca de Python dedicada a la Data Science”. Formación en ciencia de datos | DataScientest.com. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://datascientest.com/es/pandas-python

[22] “Openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.1.3 documentation”. openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files — openpyxl 3.1.3 documentation. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://openpyxl.readthedocs.io/en/stable/

[23] “GitHub - google-gemini/generative-ai-python: The official Python library for the Google Gemini API”. GitHub. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://github.com/google-gemini/generative-ai-python

[24] “W3Schools.com”. W3Schools Online Web Tutorials. Accedido el 6 de agosto de 2024. [En línea].
Disponible: https://www.w3schools.com/python/python_json.asp

[25] “Gemini: Preguntas frecuentes”. Google Gemini. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://gemini.google.com/faq?hl=es-419

[26] Y. Fernández. “Google Gemini: Qué es, cómo funciona, diferencias con GPT y cuándo podrás usar este modelo de inteligencia artificial”. Xataka - Tecnología y gadgets, móviles, informática, electrónica. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.xataka.com/basics/google-gemini-que-como-funciona-diferencias-gpt-cuando-podras-usar-este-modelo-inteligencia-artificial

[27] “Precios de la API de Gemini | Google AI for Developers”. Google AI for Developers. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://ai.google.dev/pricing?hl=es-419

[28] “Pricing | OpenAI”. OpenAI. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://openai.com/api/pricing/

[29] “¿Qué son y para qué sirven los microservicios? Beneficios de los microservicios”. Red Hat. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.redhat.com/es/topics/microservices

[30] “¿Qué es Docker y cómo funciona?” Red Hat. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://www.redhat.com/es/topics/containers/what-is-docker

[31] P. Delgado-Santos, G. Stragapede, R. Tolosana, R. Guest, F. Deravi y R. Vera-Rodriguez, “A survey of privacy vulnerabilities of mobile device sensors”, *ACM Comput. Surv. (CSUR)*, vol. 54, 2022, art. n.º 224. Accedido el 5 de agosto de 2024. [En línea]. Disponible: https://doi.org/10.1145/3510579

[32] European Union Agency for Cybersecurity (EU body or agency), “Privacy and data protection in mobile applications”, *Eur. Union*, 2017. Accedido el 5 de agosto de 2024. [En línea].
Disponible: https://data.europa.eu/doi/10.2824/114584

[33] “Empresas aseguradoras”. COSEDE. Accedido el 6 de agosto de 2024. [En línea]. Disponible: https://www.cosede.gob.ec/wp-content/uploads/2022/03/Empresas-aseguradoras.docx

[34] “Listado de compañías extranjeras”. Superintendencia de compañías, valores y seguros. Accedido el 6 de agosto de 2024. [En línea].
Disponible: https://mercadodevalores.supercias.gob.ec/reportes/companiasExtranjeras.jsf

[35] Y.-A. de Montjoye, C. Hidalgo, M. Verleysen and V. Blondel, “Unique in the Crowd:

The Privacy Bounds of Human Mobility," 2013.

[36] J. Achara, G. Acs and C. Castelluccia, "On the Unicity of Smartphone Applications," in 14th ACM CCS Workshop on Privacy in Electronic Society (ACM WPES), 2015.

[37] "Datos de usuario". Google, Accedido el 15 de agosto de 2024. [En línea]. Disponible: https://support.google.com/googleplay/android-developer/answer/10144311, 2024.

[38] "Proporciona información para la sección de Seguridad de los datos de Google Play". Google, Accedido el 16 de agosto de 2024. [En línea]. Disponible: https://support.google.com/googleplay/android-developer/answer/10787469 ?hl=es-419, 2024.

6 ANEXOS

En esta sección se presenta un resumen de los anexos que se encuentran en el presente trabajo de titulación:

ANEXO A. Lista de información que contiene la política de privacidad.

ANEXO B. Lista de permisos sensibles.

ANEXO C. Lista de aplicaciones solicitadas a UPM

ANEXO D. Diagrama de despliegue de microservicio del primer módulo.

ANEXO E. Enlace con el resultado del microservicio.

ANEXO F. Conjunto de datos con las etiquetas de privacidad.

ANEXO G. Conjunto de datos con los metadatos de las aplicaciones.

ANEXO H. Enlace con el resultado final.

ANEXO I. Repositorio GitHub con los módulos.

ANEXO A

Lista de información que contiene la política de privacidad.

Contenido de la política	Definición
Controlador	Identificador de las organizaciones que recopilan datos personales y sus datos de contacto.
Datos	Tipos de datos personales utilizados para alguna operación de procesamiento.
Operaciones	Operaciones de procesamiento que se realizan sobre los datos personales, como recopilación, organización, almacenamiento, divulgación, transmisión, etc. Estas operaciones pueden ser realizadas por el propio controlador o por terceros.
Propósito	El objetivo comercial detrás del procesamiento que se realizan sobre los datos personales.
Consentimiento	Los usuarios deben tener el poder de decidir si dar o no sus datos personales para las operaciones de datos descritos en la política de privacidad.
Acceso	Información sobre los derechos de los sujetos de los datos para acceder, editar y eliminar sus datos personales una vez que han sido recopilados y cómo hacerlos valer.
Retención	Cantidad de tiempo que la organización o terceras partes conservarían los datos personales.
Seguridad	Medidas tomadas para proteger los datos personales.
Cambios	Detalles acerca de cómo se va a comunicar a los usuarios de los cambios de la política de privacidad.
Niños	Aspectos relacionados con el procesamiento de datos de los niños, ya que estos son considerados como individuos vulnerables, por lo que el tratamiento de sus datos debe requerir de información adicional.

ANEXO B

Lista de permisos sensibles.

No.	Tipo de dato personal	Descripción	Nombre del permiso	Nivel de protección
1	Mensajes y llamadas	SMS, MMS, llamadas	RECEIVE_SMS	Peligrosa
2			READ_SMS	Peligrosa
3			RECEIVE_MMS	Peligrosa
4			PROCESS_OUTGOING_CALLS	Peligrosa
5			CALL_PHONE	Peligrosa
6			ANSWER_PHONE_CALLS	Peligrosa
7	Audio y video	Multimedia	CAMERA	Peligrosa
8			RECORD_AUDIO	Peligrosa
9			ADD_VOICEMAIL	Peligrosa
10	Sensores	Acceso a sensores	BODY_SENSORS	Peligrosa
11			BODY_SENSORS_BACKGROUND	Peligrosa
12	Localización	Varios permisos acerca de localización de usuarios	ACCESS_COARSE_LOCATION	Peligrosa
13			ACCESS_FINE_LOCATION	Peligrosa
14			ACCESS_BACKGROUND_LOCATION	Peligrosa
15			ACCESS_LOCATION_EXTRA_COMMANDS	Normal
16			ACCESS_MEDIA_LOCATION	Peligrosa
17	Almacenamiento externo	Almacenamiento	WRITE_EXTERNAL_STORAGE	Peligrosa
18			READ_EXTERNAL_STORAGE	Peligrosa
19	Contacto	Lista de contactos	READ_CONTACTS	Peligrosa
20			WRITE_CONTACTS	Peligrosa
21	Uso/historial	Registros de llamadas	READ_CALL_LOG	Peligrosa
22			WRITE_CALL_LOG	Peligrosa
23	Identificadores	Cuentas	GET_ACCOUNTS	Peligrosa
24	Credenciales	Métodos de autenticación	USE_BIOMETRIC	Normal
25			USE_FINGERPRINT	Normal

ANEXO C

LISTA DE APLICACIONES SOLICITADAS A UPM

ANEXO D

Diagrama de despliegue de microservicio del primer módulo.

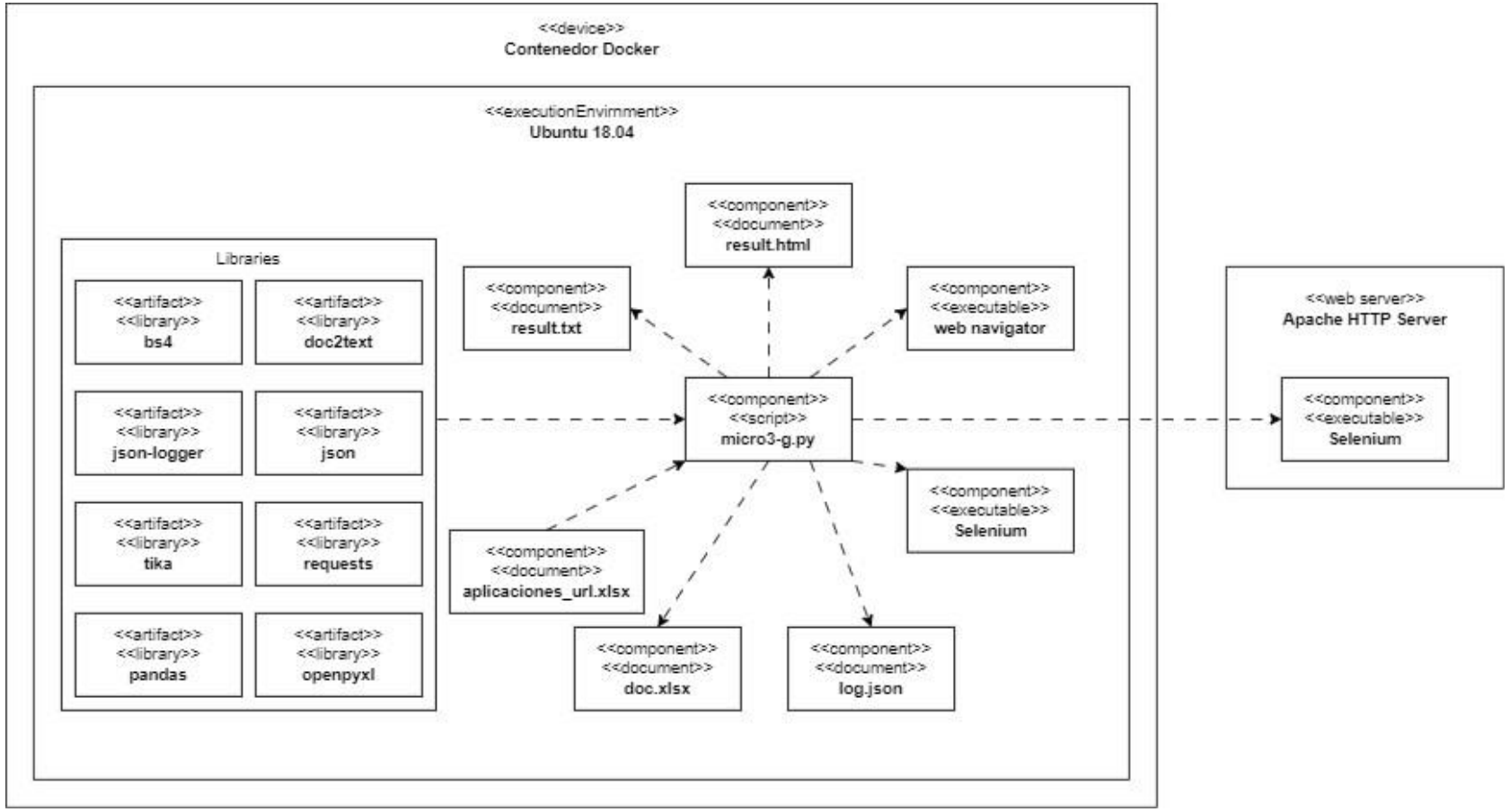

Figura 2.14 Diagrama de despliegue del microservicio del primer módulo.

ANEXO E

RESULTADO DEL MICROSERVICIO

ANEXO F

CONJUNTO DE DATOS CON LAS ETIQUETAS DE PRIVACIDAD

ANEXO G

CONJUNTO DE DATOS CON LOS METADATOS DE LAS APLICACIONES

ANEXO H

RESULTADOS FINALES

ANEXO I

REPOSITORIO GITHUB CON EL CÓDIGO DE LOS MÓDULOS