

Desarrollo de una herramienta para el control de la privacidad en Android

Memoria del Trabajo de Fin de Grado

presentada para optar al Grado de

Ingeniería Informática de Gestión y sistemas de Información

por

Iñaki Sánchez Jiménez

Director: Pablo González Nalda

28 de septiembre de 2020

Copyright ©2020 IÑAKI SÁNCHEZ JIMÉNEZ. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Agradecimientos

Gracias a mis padres, que siempre han estado a mi lado para apoyarme en mis decisiones y ayudarme en todo, sin excepciones. Gracias a vosotros soy quien soy, y sin vosotros este camino habría sido mucho más duro.

Gracias a Alvaro, Imanol, Josu y Keoma, grandes amigos que siempre han sabido darme consejos para ser mejor persona, y siempre han estado en los momentos más duros para ayudarme a levantarme.

Gracias a todos los amigos que he hecho en la carrera y que de una manera u otra siguen formando parte de mi vida: Borja, Eneko, Eztizen, Jon, Leire, Mishelle y Raúl. Cada asignatura habría sido muy diferente sin vuestras aportaciones, que han hecho de ella una experiencia única. También gracias a los amigos que he hecho gracias a la carrera aunque no hayamos compartido parte de ella: Alejandro, Borja, Iker, Pablo y Raul. Sin vosotros esos pequeños momentos no solo aprendiendo cosas diferentes, si no disfrutando de la informática de otra manera, nunca habrían sido posible. Gracias por abrirme los ojos ante muchas cosas que si no fuese por vosotros, nunca habría conocido.

Gracias a dos grandes amigos que hice durante la carrera aunque no coincidiese en clase con ellos prácticamente: Ander y Joseba. Con vosotros he vivido una pequeña parte de mi vida que no solo me ha valido profesionalmente, si no personalmente.

Gracias a los profesores de la carrera que me han enseñado, y a aquellos que aunque no fuesen mis profesores, no les ha importado enseñarme algo. Gracias por satisfacer una curiosidad que me ha permitido no tener miedo de no saber algo, y alentar las ganas de llenar ese desconocimiento con conocimiento. Gracias Pablo por el esfuerzo de ayudarme con este TFG, que sin quererlo me ha llevado mucho más tiempo del esperado.

Por último, muchas gracias a Oihane, quien se ha convertido en una fuente de apoyo indispensable y quien me alienta a hacer todo lo que me proponga.

Índice general

Agradecimientos.....	V
I. Alcance del trabajo	1
1. Descripción, Objetivos y Motivación	1
1.1. Descripción.....	1
1.2. Objetivos	2
1.3. Motivación	3
2. Resumen y organización de la memoria.....	4
3. Viabilidad	5
3.1. Requisitos funcionales	5
3.2. Planificación del tiempo.....	6
3.2.1. Fase 1	6
3.2.2. Fase 2	6
3.2.3. Tareas.....	7
3.2.4. Gestión de costos	10
3.2.5. Presupuesto	11
3.3. Gestión de riesgos.....	13
3.3.1. Explicación y plan de contingencia.....	13
II. Fase I: Estado del arte de la privacidad en dispositivos inteligentes.....	16
4. Introducción.....	16
5. Conceptos generales	17
5.1. Información recolectable	18
5.1.1. Información recolectable mediante hardware	18
5.1.2. Información recolectable mediante software.....	18
5.2. Uso de la información	19
5.3. Seguridad de la información	20
5.3.1. Información comprometida	20
6. Recolección de datos en sistemas operativos	21
6.1. Sistemas de uso personal.....	22
6.1.1. Windows	22
6.1.2. Apple	23
6.1.3. Google	24
6.1.4. Ubuntu	25
6.2. Compañías de software.....	26
6.2.1. Facebook	26
6.2.2. Amazon	27
6.3. Compañías de hardware	29
6.3.1. Samsung	29
6.3.2. Huawei	30

7. Conclusiones	32
7.1. La privacidad en los dispositivos móviles Android	34
7.2. Medidas preventivas	35
III. Fase 2: Desarrollo de la aplicación	37
8. Introducción	37
9. Tecnologías y herramientas	38
9.1. Android	38
9.1.1. Particiones	38
9.1.2. Custom recovery	39
9.1.3. Magisk	39
9.1.4. Android Debug Bridge	40
9.1.5. Android Shell	40
9.1.6. Android Package Manager	40
9.2. Python	41
9.3. Git	42
10. Desarrollo de la aplicación	43
10.1. Arquitectura	43
10.2. Conexión con el dispositivo	44
10.3. Interfaz grafica	47
10.4. Navegación	48
10.4.1. Listado de paquetes	48
10.4.2. Desinstalación de paquetes	49
10.5. Conclusiones	50
11. Implementación de la aplicación	51
11.1. Estructura del proyecto	51
11.2. Integración de ADB	54
11.2.1. Descarga	54
11.2.2. Interacciones	55
11.3. Uso de ADB	56
11.3.1. Lectura de paquetes	56
11.3.2. Modificación de paquetes	56
11.4. Interfaz gráfica	58
IV. Análisis y conclusiones del trabajo	62
12. Conclusiones	62
12.1. Problemas durante el desarrollo	63
12.2. Trabajo futuro	64
V. Apéndices	66
Código fuente y documentación del proyecto	66

Bibliografía.....	67
GNU Free Documentation License	69
0. PREAMBLE.....	69
1. APPLICABILITY AND DEFINITIONS	69
2. VERBATIM COPYING.....	70
3. COPYING IN QUANTITY	71
4. MODIFICATIONS.....	71
5. COMBINING DOCUMENTS	73
6. COLLECTIONS OF DOCUMENTS.....	73
7. AGGREGATION WITH INDEPENDENT WORKS.....	73
8. TRANSLATION.....	74
9. TERMINATION	74
10. FUTURE REVISIONS OF THIS LICENSE	74
11. RELICENSING	75

Índice de ilustraciones

<i>Ilustración 6.1 Logos de sistemas operativos famosos</i>	21
<i>Ilustración 6.2 Logo de Windows 10</i>	23
<i>Ilustración 6.3 Logo de iOS</i>	24
<i>Ilustración 6.4 Logo de Android</i>	25
<i>Ilustración 6.5 Logo de Ubuntu</i>	25
<i>Ilustración 6.6 Logo de Facebook</i>	27
<i>Ilustración 6.7 Logo de Amazon</i>	28
<i>Ilustración 6.8 Logo de Samsung</i>	30
<i>Ilustración 6.9 Logo de Huawei</i>	31
<i>Ilustración 10.1 Representación de la arquitectura hexagonal</i>	43
<i>Ilustración 10.2 Activación de las opciones de desarrollo</i>	44
<i>Ilustración 10.3 Activación del modo debug</i>	45
<i>Ilustración 10.4 Ventana para permitir acceso del ordenador al dispositivo</i>	46
<i>Ilustración 10.5 Ejemplo de distribución de filas y columnas en Tkinter</i>	47
<i>Ilustración 10.6 Botón de listar paquetes de la aplicación</i>	48
<i>Ilustración 10.7 Botón de desinstalar paquetes de la aplicación</i>	49
<i>Ilustración 11.1 Estructura del módulo principal del proyecto</i>	52
<i>Ilustración 11.2 Estructura general del proyecto</i>	53
<i>Ilustración 11.3 Ventana principal de la aplicación al ejecutarla</i>	59
<i>Ilustración 11.4 Ventana de la aplicación una vez listados los paquetes</i>	60
<i>Ilustración 11.5 Ventana de la aplicación al filtrar los paquetes por un texto</i>	60

Índice de tablas

<i>Tabla 3.1 Recursos materiales físicos</i>	<i>10</i>
<i>Tabla 3.2 Recursos materiales digitales</i>	<i>11</i>
<i>Tabla 3.3 Recursos amortizables</i>	<i>11</i>
<i>Tabla 3.4 Recursos específicos del proyecto.....</i>	<i>11</i>
<i>Tabla 3.5 Recursos humanos.....</i>	<i>11</i>
<i>Tabla 3.6 Amortizaciones de hardware y software</i>	<i>12</i>
<i>Tabla 3.7 Total presupuesto</i>	<i>12</i>
<i>Tabla 3.8 Enumeración de riesgos del proyecto</i>	<i>13</i>

Índice de código fuente

<i>Código fuente 11.1 Función de descarga de ADB</i>	<i>54</i>
<i>Código fuente 11.2 Función de descompresión de la descarga de ADB</i>	<i>54</i>
<i>Código fuente 11.3 Función de creación de directorio para archivos descargados.....</i>	<i>54</i>
<i>Código fuente 11.4 Función para eliminar archivo ADB descargado</i>	<i>55</i>
<i>Código fuente 11.5 Comandos para entornos Windows</i>	<i>55</i>
<i>Código fuente 11.6 Comandos para entornos Linux.....</i>	<i>55</i>
<i>Código fuente 11.7 Comandos para entornos Darwin</i>	<i>55</i>
<i>Código fuente 11.8 Función para listar paquetes del dispositivo</i>	<i>56</i>
<i>Código fuente 11.9 Función para desinstalar un paquete del dispositivo</i>	<i>57</i>
<i>Código fuente 11.10 Código para añadir entrada de texto y botón de filtrar</i>	<i>58</i>
<i>Código fuente 11.11 Código para añadir columnas con diferente información de los paquetes</i>	<i>58</i>
<i>Código fuente 11.12 Código para añadir botón de listado de paquetes y desinstalación de un paquete..</i>	<i>59</i>

I. Alcance del trabajo

1. Descripción, Objetivos y Motivación

1.1. Descripción

A lo largo de este Trabajo de Fin de Grado se desarrolla un estudio sobre el campo de la privacidad en los dispositivos que cualquier persona puede usar en su día a día, ya sea en dispositivos personales, dispositivos para el desarrollo de un trabajo o dispositivos que usamos por el uso de servicios como el transporte público. En base a todo ese estudio se desarrolla una aplicación para dispositivos móviles Android que busca que cualquier persona, sin conocimientos avanzados, pueda decidir qué paquetes hay presentes en su dispositivo móvil Android. Esta herramienta se basa en una pequeña funcionalidad que, de no tener ciertos conocimientos avanzados en la obtención de un acceso privilegiado dentro del dispositivo, sería imposible de realizar.

Cabe destacar que esta funcionalidad se basa en ciertas herramientas [1] que el propio sistema Android provee al usuario final que, aunque estén al principio bloqueadas y su uso esté pensado para los proveedores finales de terminales basados en Android, pueden aprovecharse para tomar cierto control sobre el dispositivo.

Además del desbloqueo de estas funcionalidades por parte del usuario, la aplicación necesita ciertas librerías [2] que los responsables del desarrollo de Android ponen a disposición de cualquier persona que quiera usarlas. Aun así, el uso de estas librerías está orientada a usuarios avanzados ya que se necesitan conocimientos en el uso de terminales.

1.2. Objetivos

El objetivo principal de este proyecto es el de divulgar de una manera amigable lo que se esconde detrás de cualquier sistema, aplicación e incluso servicio que consumimos en nuestro día a día. Se quiere hacer hincapié en cómo, sin que los usuarios seamos totalmente conscientes de ello, toda nuestra actividad es de alguna manera recolectada y analizada.

El segundo objetivo consiste en implementar una aplicación la que cualquier persona pueda elegir qué paquetes hay ejecutándose en su dispositivo Android, ya que esto no es posible gestionarlo desde el terminal. Esto es debido a que ciertos paquetes tienen restringida su modificación e incluso desinstalación, lo que no tiene sentido para aquellos usuarios que decidan no hacer uso de esas características.

Como último objetivo, se establece ofrecer una aplicación de código abierto para que cualquier persona pueda ver el código fuente de la misma y haga mejoras o avance en su desarrollo, haciendo que sea el usuario final el que disfrute de estos avances ofrecidos por terceras personas.

1.3. Motivación

Hoy en día todas las personas estamos rodeadas de dispositivos inteligentes. Ya sea a través de un dispositivo como un teléfono móvil, tablet, ordenador, o incluso a través del consumo de servicios como compras online, transporte público o incluso la sanidad, tenemos a nuestra disposición todo tipo de acciones que podemos realizar haciendo uso de estos dispositivos.

Todas estas posibilidades que se ponen a nuestro alcance tienen una cosa en común: cada interacción con estas aplicaciones o servicios es grabada. Dejamos una huella, con una identidad, una marca de tiempo, y una consecuencia a nuestra acción. Esta información es muy valiosa para los proveedores, ya que, bajo el pretexto de una experiencia de usuario final extraordinaria, han pasado a monitorizar cada una de nuestras acciones.

En este trabajo se trata de concienciar en cierta manera de que somos los usuarios quienes debemos de gestionar y ser conscientes sobre qué información compartimos y de qué condiciones aceptamos al ceder esta información.

2. Resumen y organización de la memoria

Esta memoria consta de cinco partes, en las que se ha realizado una división para presentar las diferentes partes que tiene el proyecto.

En la primera parte, denominada Alcance del Trabajo, se habla sobre la viabilidad de un *MVP* (*Minimum Viable Product*¹, traducido en castellano Producto Mínimo Viable) y los requisitos para desarrollarlo. Se trata como se ha planificado y llevado a cabo el proyecto siguiendo metodologías ágiles.

En la segunda parte, relacionada con el estado del arte de la privacidad de los usuarios, se analiza sobre el estado actual de las configuraciones de privacidad que ofrecen por defecto los sistemas más usados, viendo qué tipo de información recogen.

En la tercera parte se desarrolla la aplicación para Android, centrándose completamente en este sistema operativo, que ayuda a los usuarios finales a tener un mayor control sobre qué paquetes hay ejecutándose en su dispositivo.

En la cuarta parte, *Análisis y conclusiones del trabajo*, se describen sobre los problemas encontrados durante el desarrollo de este trabajo, así como futuros desarrollos posibles a la aplicación implementada.

En la última parte del trabajo, con el nombre *Apéndices*, se detalla la bibliografía utilizada y las referencias al contenido sobre el que se han basado las ideas desarrolladas y las conclusiones a las que se ha llegado.

¹ https://es.wikipedia.org/wiki/Producto_viable_m%C3%ADnimo

3. Viabilidad

Este capítulo tiene como objetivo elaborar un estudio para comprobar la viabilidad del proyecto, evaluando los posibles gastos y los posibles riesgos asociados a este, que podrían alargar la duración del proyecto o dificultar parte del desarrollo.

3.1. Requisitos funcionales

Los requisitos funcionales de la aplicación se elaboran en base a los objetivos descritos y que son los siguientes:

- Elaborar un estado del arte que analice el campo de la privacidad, su impacto y cómo podemos mejorar nuestra privacidad en un mundo cada vez más digital.
- Elaborar una aplicación que permita la obtención de información de los paquetes presentes en dispositivos Android, sin necesidad de conocimientos técnicos.
- Elaborar una aplicación que permita la desinstalación de paquetes en dispositivos Android, sin necesidad de conocimientos técnicos.

3.2. Planificación del tiempo

3.2.1. Fase 1

En la Fase 1 simplemente se ha creado una base de conocimiento, ya que de alguna manera es también una justificación para hacer la aplicación y de la misma forma la fuente de información que se quiere divulgar.

Se ha intentado seguir un orden a la hora de analizar toda la información disponible para que cada paso previo ayudase en los posteriores:

- Investigar cómo tratan y recolectan los sistemas operativos la información de los usuarios finales, que además son la base sobre la que se ejecuta cualquier aplicación que finalmente usan los usuarios
- Investigar cómo las aplicaciones recolectan información de los usuarios, además de sus posibles usos
- Investigar cómo afecta esta recolección de información a los usuarios finales y explicar por qué seguridad y privacidad deben ir siempre de la mano

Una vez analizada esta información, se ha trabajado en la justificación e importancia que puede tener un control sobre las aplicaciones que los usuarios finales creen tener pero que muchas veces no se cumple.

Es difícil además hacer un cálculo del tiempo invertido y del coste del mismo, porque, aunque haya sido el primer paso para realizar la aplicación, en realidad, en ningún momento se ha dejado de buscar nueva información ni de incrementar esa base de conocimiento.

Como se ve un poco en la Fase 2, podemos decir que esta primera fase también es un tanto “ágil” ya que un poco de conocimiento basta para empezar todo el proceso de implementación, pero no por ello termina cuando empieza la siguiente, sino que ambas son complementarias.

3.2.2. Fase 2

Para la Fase 2 se ha decidido usar Scrum² como base para el desarrollo funcional de la aplicación. Mediante esta metodología ágil, vamos a poder definir un número de tareas a realizar en un tiempo de 2 semanas, incluyendo días naturales, lo que nos deja 10 días de trabajo completo.

Además, gracias a que se siguen metodologías ágiles, no se necesita que el desarrollo completo de la aplicación esté descompuesto en tareas, sino que se pueden definir las primeras para empezar el desarrollo, e ir creando nuevas según vaya surgiendo la necesidad.

Para esos 10 días de trabajo disponibles, esta metodología se puede definir en varios pasos a la hora de trabajar, que se definen más adelante:

- Crear tareas
- Reuniones para asignar peso a las tareas (también conocidas como historias de usuario)
- Reuniones para valorar como ha ido el Sprint y si se han cumplido los objetivos
- Reuniones para organizar el Sprint
- Trabajar en las tareas

² <https://proyectosagiles.org/que-es-scrum/>

Para rellenar estos 10 días de trabajo se tendrán en cuenta las horas disponibles, pero al contrario que con la tradicional gestión de proyectos en la que cada tarea tiene un número de horas definidas, aquí cada tarea va a tener un peso asignado.

Este peso se elige entre todos los miembros del equipo, que también habrá elegido el número de pesos asignables y el valor de cada uno. Para este caso se ha decidido usar 3 tallas de ropa: S, M y L. Como en Scrum las tareas necesitan un número cardinal, asignamos uno para cada uno de ellos:

- S tendrá un peso de 1
- M tendrá un peso de 3
- L tendrá un peso de 5

La manera de valorar el peso a asignar a una tarea tampoco se reflejará en el tiempo que esta necesita, si no en la dificultad de realizar esa tarea (que puede ser añadir una nueva funcionalidad o arreglar una funcionalidad que no sea correcta).

Para comenzar un sprint, simplemente se añadirán diferentes tareas al sprint, hasta obtener más o menos el peso total deseado del sprint.

Si por ejemplo estimamos que la capacidad de un sprint es de 10 puntos, podemos quedarnos algo por debajo o algo por encima, ya que no es un número objetivo si no una mera referencia de la capacidad posible del equipo.

3.2.3. Tareas

Vamos a tener tres tipos de tareas:

- Task: usaremos este tipo de tarea para aquellas que no aportan valor al producto
- Story: usaremos este tipo de tarea para aquellas que aportan valor al producto
- Bug: usaremos este tipo de tarea para representar problemas o funcionamientos no deseados de la aplicación

En base a esto, se han definido las siguientes tareas:

Referencia	APU-1
Tipo	Task
Nombre	Buscar información sobre ADB
Descripción	Leer la documentación de Android para ver qué acciones están disponibles para usar con ADB y que limitaciones se pueden encontrar a la hora de usarlo.
Peso	M (3)

Referencia	APU-2
Tipo	Task
Nombre	Empezar proyecto en Python 3.8
Descripción	Comenzar proyecto creando la estructura a seguir más adelante, definiendo las diferentes capas y módulos.
Peso	S (1)

Referencia	APU-3
Tipo	Task
Nombre	Investigar cómo gestionar el proyecto con Virtual Env
Descripción	Investigar cómo se puede usar virtual env para que la gestión de las dependencias sea automática.
Peso	M (3)

Referencia	APU-4
Tipo	Task
Nombre	Investigar qué framework usar para la GUI
Descripción	Investigar los frameworks disponibles para usar con Python para crear de una manera sencilla una GUI, y elegir uno.
Peso	L (5)

Referencia	APU-5
Tipo	Story
Nombre	Crear ventana con TKinter
Descripción	Crear una ventana usando el framework para Python Tkinter. que no tenga nada y que se inicie al ejecutar el programa.
Peso	M (3)

Referencia	APU-6
Tipo	Task
Nombre	Crear tarea de descarga de ADB en la aplicación
Descripción	Crear una tarea que al iniciar la aplicación descargue (si no están ya presentes) los archivos necesarios para poder usarlos.
Peso	L (5)

Referencia	APU-7
Tipo	Task
Nombre	Hacer que la tarea de descarga funcione en diferentes entornos
Descripción	Teniendo ya una tarea de descarga programada, hacerla compatible con diferentes entornos. Esto requiere la descarga de diferentes archivos según el entorno, que tendrá que comprobarse para la posterior descarga.
Peso	M (3)

Referencia	APU-8
Tipo	Story
Nombre	Obtener información de los paquetes instalados en el dispositivo
Descripción	Mediante la interacción con los paquetes de ADB, conseguir una comunicación con el dispositivo para conseguir información de los paquetes instalados.
Peso	M (3)

Referencia	APU-9
Tipo	Story
Nombre	Enseñar en la GUI la información obtenida del dispositivo
Descripción	Mostrar en la aplicación la información obtenida referente a los paquetes instalados en la ventana creada.
Peso	M (3)

Referencia	APU-10
Tipo	Story
Nombre	Crear botón en la GUI para obtener información del dispositivo
Descripción	Hacer que la obtención de la información del dispositivo sea mediante un botón que presione el usuario y mostrarla.
Peso	S (1)

Referencia	APU-11
Tipo	Story
Nombre	Crear columna para enseñar ruta de los paquetes obtenida del dispositivo
Descripción	Crear una columna que muestre simplemente la ruta en la que se encuentras los paquetes que tiene el dispositivo.
Peso	M (3)

Referencia	APU-12
Tipo	Story
Nombre	Crear columna para enseñar el nombre de los paquetes obtenida del dispositivo
Descripción	Crear una nueva columna junto a la de la ruta del paquete que muestre el nombre del paquete instalado.
Peso	S (1)

Referencia	APU-13
Tipo	Story
Nombre	Crear columna para enseñar el id del paquete que se instala en el dispositivo
Descripción	Crear una nueva columna junto a la del nombre del paquete que muestre el id del paquete instalado.
Peso	S (1)

Referencia	APU-14
Tipo	Story
Nombre	Crear botón para filtrar texto en la información de las apps obtenida
Descripción	Crear un botón en la GUI que busque en cualquiera de las columnas el texto introducido por el usuario.
Peso	M (3)

Referencia	APU-15
Tipo	Bug
Nombre	Hacer que el botón de filtrar no haga nada cuando tenga un texto vacío
Descripción	Teniendo el cuadro de texto para introducir un filtrado vacío, si se busca con un texto vacío, devuelve el listado completo de paquetes de nuevo. Debería de dejar el listado tal y como estaba antes de intentar filtrar.
Peso	S (1)

Referencia	APU-16
Tipo	Task
Nombre	Abstraer la funcionalidad para que funcione en diferentes dispositivos
Descripción	Hacer que la aplicación funcione indiferentemente desde el entorno donde se ejecute.
Peso	L (5)

Referencia	APU-17
Tipo	Story
Nombre	Crear funcionalidad para desinstalar paquetes
Descripción	Añadir la funcionalidad para poder desinstalar paquetes desde la aplicación, haciendo que sea posible desde todos los entornos compatibles.
Peso	L (5)

Referencia	APU-18
Tipo	Story
Nombre	Crear botón que desinstale un paquete elegido por el usuario
Descripción	Crear un botón en la GUI que deje al usuario elegir un paquete del listado, y desinstalar el mismo.
Peso	M (3)

3.2.4. Gestión de costos

En esta sección se hace un análisis de los costos de los recursos que van a usarse para llevar a cabo todo el proyecto. Debemos además diferenciar entre los recursos físicos de los digitales, siendo estos últimos los que usaremos mediante los recursos físicos.

En la siguiente tabla identificamos los recursos físicos:

Recurso físico	Tipo	Coste variable	Coste fijo
Iñaki Sánchez	Trabajador	14 €/h	0 €
Portátil	Material	0 €/h	1499 €
Huawei P Smart Plus	Material	0 €/h	215 €

Tabla 3.1 Recursos materiales físicos

En la siguiente tabla identificamos los recursos digitales:

Recurso digital	Coste variable	Coste fijo
Windows 10	0 €/h	0 € ³
macOS 10.15	0 €/h	0 €
Ubuntu 20.04 LTS	0 €/h	0 €
Atlassian Jira Cloud	0 €/h	0 €
Microsoft Office 2019	0 €/h	150 €
PyCharm Community	0 €/h	0 €
GitHub	0 €/h	0 €
Git	0 €/h	0 €
Android Debug Bridge	0 €/h	0 €

Tabla 3.2 Recursos materiales digitales

3.2.5. Presupuesto

En este análisis de los costes del proyecto vamos a detallar los costes reales que tiene el proyecto teniendo en cuenta que no todo el material usado es único para el proyecto, si no que tiene un periodo de amortización y por lo tanto el coste del mismo depende del uso que haga el proyecto del recurso.

Para el caso específico de este proyecto, podemos identificar como gasto único para el proyecto el trabajador que va a desarrollar la aplicación y que tiene un coste dependiendo de las horas que invierta para crear la aplicación.

En cambio, en el caso de recursos que se amortizan en el tiempo tenemos el portátil con el que el trabajador va a desarrollar la aplicación, teniendo este un tiempo de amortización de 3 años (3 años * 200 días laborables al año * 8 horas = 4800 horas). También se amortiza de la misma manera el software Microsoft Office que saca una versión nueva cada 3 años.

Recurso	Coste	Tiempo de amortización
Portátil	0,32 €/h	4800 h
Microsoft Office 2019	0,03 €/h	4800 h

Tabla 3.3 Recursos amortizables

En cuanto al dispositivo móvil, ya que es un recurso específico que requiere el proyecto y que posteriormente no se va a usar, se incluye como gasto directo.

Recurso	Coste	Tiempo de amortización
Huawei P Smart Plus	215 €	-

Tabla 3.4 Recursos específicos del proyecto

Por último, se tiene en cuenta el gasto del trabajador, que invierte 285h para crear la aplicación.

Recurso	Coste	Tiempo de uso	Total
Iñaki Sánchez	14 €/h	285 h	3990 €

Tabla 3.5 Recursos humanos

³ <https://insider.windows.com/en-us/>

Unificando todos los costes de hardware y software, tenemos:

Recurso	Coste	Tiempo de uso	Importe
Portátil	0,32 €/h	285 h	91,2 €
Huawei P Smart Plus	215 €	40 h	215 €
Windows 10	0 €	105 h	0 €
macOS 10.15	0 €	10 h	0 €
Ubuntu 20.04 LTS	0 €	10 h	0 €
Atlassian Jira Cloud	0 €	100 h	0 €
Microsoft Office 2019	0,03 €/h	40 h	1,2 €
PyCharm Community	0 €	120 h	0 €
GitHub	0 €	150 h	0 €
Git	0 €	285 h	0 €
Android Debug Bridge	0 €	40 h	0 €
TOTAL			307,4 €

Tabla 3.6 Amortizaciones de hardware y software

Teniendo, finalmente, un presupuesto total:

Concepto	Importe
Recursos de Trabajo	3990 €
Recursos Materiales	215 €
Costo fijo	0 €
Amortizaciones	92,4 €
SUMA	4297,4 €
Gastos generales (10%)	429,74 €
Beneficio (20 %)	859,48 €
SUBTOTAL	5586,62 €
IVA (21%)	1173,20 €
TOTAL	6759,82 €

Tabla 3.7 Total presupuesto

Concluyendo entonces que el coste total del proyecto suma la cantidad de seis mil setecientos cincuenta y nueve con ochenta y dos euros.

3.3. Gestión de riesgos

En este apartado se evalúan posibles riesgos y amenazas que pueden llegar a provocar retrasos en el desarrollo del proyecto. Por ello se han identificado los diferentes riesgos que pueden ir asociados al proyecto y que pueden ocurrir en cualquier momento.

En la siguiente tabla se muestran los riesgos identificados:

Riesgo	Peligrosidad
Perdida de información	Alta
Enfermedades	Alta
Dificultades en la implementación de la fase 2	Media
Problemas en los recursos materiales	Baja

Tabla 3.8 Enumeración de riesgos del proyecto

3.3.1. Explicación y plan de contingencia

Tras la identificación de los riesgos asociados y presentes en la tabla [t], vamos a idear un plan de contingencia que busca dejar definidas acciones preventivas para que riesgo, para que en caso de que ocurran, el impacto sea el menor posible.

Riesgo	Pérdida de información
Descripción	Pérdida de la información recolectada para cualquiera de las fases, lo que haría no tener una base de conocimiento.
Probabilidad	Baja
Peligrosidad	Alta
Medidas preventivas	Uso de software que sincroniza en la nube todo lo aportado, como Google Drive o GitHub.
Medidas correctoras	Recuperar la información mediante los servicios usados de manera preventiva

Riesgo	Enfermedades
Descripción	Al contraer el único trabajador una enfermedad que le incapacita, podría desembocar en retrasos e incluso la cancelación del proyecto.
Probabilidad	Baja
Peligrosidad	Alta
Medidas preventivas	Ninguna
Medidas correctoras	Ninguna

Riesgo	Dificultades en la implementación de la fase 2
Descripción	Falta de conocimiento a la hora de desarrollar la aplicación, que puede desembocar en tareas no previstas y retrasos en las previstas.
Probabilidad	Media
Peligrosidad	Alta
Medidas preventivas	Investigación completa tanto de las tareas como de las tecnologías con las que se trabaja para conocer las limitaciones
Medidas correctoras	Replanificación de tareas

Riesgo	Problemas en los recursos materiales
Descripción	Corrupción del sistema sobre el que se está desarrollando la aplicación, o pérdida del mismo por culpa de factores físicos.
Probabilidad	Baja
Peligrosidad	Baja
Medidas preventivas	Usar software que permita la reproducción completa del entorno de desarrollo, sin que este dependa ni de dispositivos físicos ni digitales.
Medidas correctoras	Reproducción completa del entorno de desarrollo

II. Fase I: Estado del arte de la privacidad en dispositivos inteligentes

4. Introducción

Esta parte de la memoria contiene información que los diferentes proveedores de sistemas han puesto al alcance de los usuarios finales, para que puedan saber qué información es recolectada y en qué momento.

En el primero, llamado *Conceptos generales*, se describen los conceptos generales, con el fin de entender mejor la privacidad de los usuarios en los sistemas que utilizan. Además, se habla de definiciones que sirven para profundizar y de cierta manera comprender completamente este campo dentro de la informática. También se analizan los efectos de que esta información sea comprometida, y como puede afectar al usuario final.

En el segundo, llamado *Recolección de datos en sistemas operativos*, se trata el impacto de las configuraciones de la privacidad que los usuarios, y como pueden utilizarse para conseguir diferentes objetivos. Además, se discute el hecho de por qué a las diferentes compañías que ofrecen software quieren recolectar la mayor cantidad de información posible y que uso pueden hacer de esta información.

Cabe destacar que el objetivo de esta fase no es solo la de obtener la información actual sobre la privacidad, sino que se busca además explicar por qué hace falta la aplicación a desarrollar en la siguiente fase. Quedará de entonces, de cierta manera, expuesto como esta aplicación intenta solventar un vacío que hoy en día ninguna aplicación cubre directamente, aunque las herramientas para hacer esto estén al alcance de todo el mundo.

De esta manera, quedara argumentada la idea principal del trabajo a desarrollar y las implicaciones que tiene no tomar ciertas medidas en los diferentes sistemas que usamos en nuestro día a día.

5. Conceptos generales

Vivimos en un mundo en el que cada día el software gana más protagonismo en el día a día de las personas. La tecnología hace mucho paso de ser un bien poco común y exclusivo de la ciencia y la investigación, para convertirse en el principal centro de ocio de las personas.

Al igual que cogemos las llaves de las puertas de casa o del coche, la cartera para llevar nuestra identificación, dinero o diferentes medios de pago, hoy en día todo el mundo sale con sus dispositivos inteligentes de casa: los teléfonos móviles.

Estos, además de ser más potentes día a día, tienen una virtud que es igual de deseada que peligrosa: la conectividad. Los teléfonos inteligentes se conectan con la nube⁴ que no para de ofrecer a sus usuarios servicios, bienes y cualquier cosa que puedan desear. Incluso poco a poco van llegando al mercado los hogares conectados, para no tener que llevar esas llaves nunca más encima. O la identificación personal para que podamos demostrar quienes somos, e incluso las tradicionales tarjetas de crédito de las que solo nosotros conocemos el pin para poder pagar.

Cada día nuestros teléfonos se vuelven más inteligentes, más capaces, y cada vez parecen más un fiel sirviente que un simple dispositivo para hacer una llamada. Hemos pasado de tener teléfonos móviles, a tener asistentes personales en la palma de nuestra mano.

Sin embargo, todo esto tiene un contratiempo: la información. A cambio de todas estas comodidades, los usuarios no dudan en ceder su información, prácticamente sea cual sea, a cambio de obtener un servicio mejor. Pero, ¿qué hay detrás de un mejor servicio? ¿Cómo puede la nube saber lo que quiero justo cuando lo quiero?

En las siguientes secciones se tratará todo esto, explicando un poco las implicaciones que tiene el uso de los dispositivos y como, cada vez que los usamos más, más saben de nosotros [15].

⁴ <https://azure.microsoft.com/es-es/overview/what-is-the-cloud/>

5.1. Información recolectable

Cuando hacemos uso de los diferentes dispositivos inteligentes, tanto los sistemas operativos como las aplicaciones pueden recolectar diferente información, pudiendo diferenciar esta entre la que se captura mediante el hardware del dispositivo al interactuar el usuario con este, y la que se captura mediante el software del dispositivo cuando el usuario lo hace uso del mismo.

Cabe destacar que esta información es enviada directamente a los dueños del servicio que se esté usando, que según la mayoría de estos anonimiza⁵ y guarda.

5.1.1. Información recolectable mediante hardware

Esta información es la que se recolecta mediante diferentes puntos de entrada que están conectados con el dispositivo. Puede ser una webcam, puede ser un micrófono (incluso el micrófono de la propia webcam) o cualquier otro dispositivo que recoja interacciones con el usuario.

Normalmente esta información es usada por un software para producir una salida o un resultado: es decir, podemos usar una webcam para desbloquear un dispositivo, dando como resultado el desbloqueo del dispositivo.

5.1.2. Información recolectable mediante software

Esta información se recolecta mediante las interacciones del usuario con el software. Puede ser desde simples clics con un ratón en determinado lugar, a reacciones a notificaciones producidas por el software que se esté usando.

Podríamos decir que este tipo de información es la recolección de la respuesta por parte de un usuario frente a posibles eventos que lanza una aplicación, aunque también se puede recoger la manera en que ese usuario usa la aplicación o donde pasa más tiempo, por ejemplo.

⁵ <https://policies.google.com/technologies/anonymization?hl=es-ES>

5.2. Uso de la información

Una vez se ha recogido la información del usuario, esta puede ser tratada de maneras muy diferentes, al igual que los fines pueden ser también muy diferentes:

- Podemos saber si un usuario es legítimo o estamos ante un Bot
- Podemos saber cómo usa un usuario nuestro software, para mejorar quizá esa sección
- Podemos saber la frecuencia con la que un usuario usa nuestra aplicación, para ofrecer más o menos servicios o condiciones especiales
- Podemos predecir futuros usos
- Podemos catalogar al usuario

Todo esto depende obviamente de la empresa que provee el software, no siendo lo mismo quienes ofrecen aplicaciones a quienes ofrecen sistemas operativos.

Si que tienen algo en común: todos buscan atraer al mayor número de usuarios posibles. Y esto se debe a que cada día la información que los usuarios comparten es cada vez más valiosa.

Cabe destacar que, toda esta información sin los correspondientes algoritmos y estadística que hay detrás, no serían más que información de uso. Pero gracias al uso de estos, las empresas son capaces de hacer predicciones sobre los usuarios que les benefician y reportan millones [16].

5.3. Seguridad de la información

Se tiende a confundir privacidad con seguridad, y viceversa. Aunque no tienen nada que ver ya que representan cosas diferentes, sí que es importante saber que la privacidad debe ir de la mano de la seguridad.

Esto es importante ya que, aunque no exponamos voluntariamente nuestros datos al software que utilicemos, este siempre nos catalogará de una manera u otra. No importa que no demos nuestro nombre o que nos inventemos uno, ya que mediante el uso que hagamos de ese software, estaremos catalogados de una manera (que depende del software que se use).

Además, es importante la seguridad porque ante un ciberataque en el que se expongan datos, si estos están correctamente guardados y seguros, aunque se exponga esa información, solo los poseedores originales de los datos serán capaces de obtener los datos originales. No solo eso, sino que da más tiempo al usuario final para que pueda modificar sus datos y evite usos fraudulentos en su nombre.

5.3.1. Información comprometida

Hay que destacar también el efecto que tiene que la información que cualquier sistema ha recolectado quede comprometida y, por lo tanto, en mayor o menor medida, expuesta a usos por parte de terceras personas.

Esta información tiene un valor incalculable para las empresas que hacen desarrollos y basan sus decisiones de negocio en estos datos, por lo que un robo de los mismos puede llegar a ser algo muy perjudicial a la empresa [21], y en parte beneficioso para las compañías que representan la competencia.

6. Recolección de datos en sistemas operativos

Todas las compañías proveedoras de software y que de algún modo recolectan información de los usuarios que las usan tienen unas políticas de privacidad propias, que el usuario debe aceptar siempre para hacer uso del servicio.

En este apartado se recoge parte de esa información disponible, recogiendo la información que estas compañías consideran útiles y con qué fines los recogen.

Se ha dividido este apartado entre las compañías que proveen software como sistemas operativos, que son las bases para usarse en distintos dispositivos, y las compañías que producen software en forma de aplicaciones, y que funcionan por encima de los sistemas operativos.

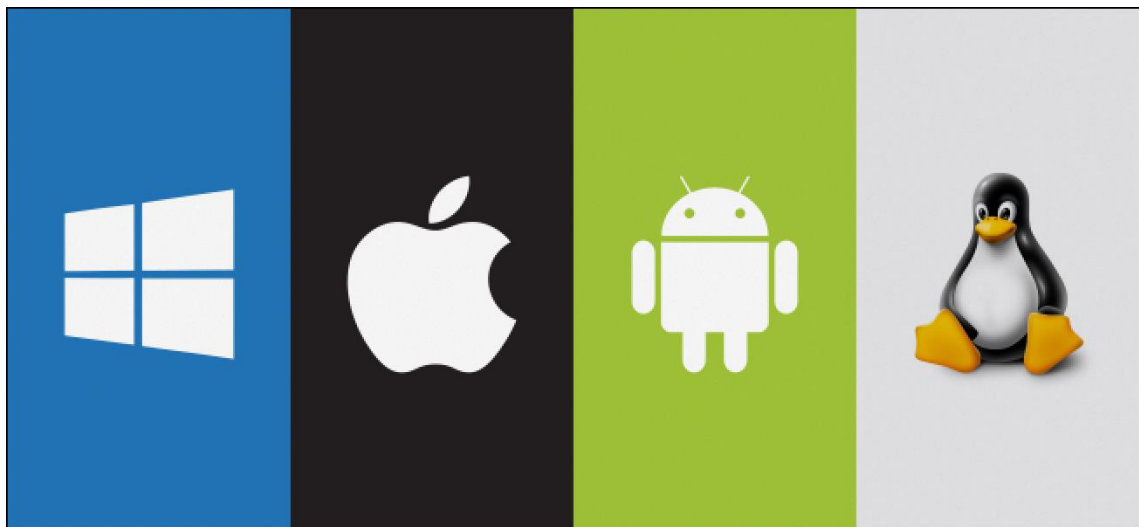


Ilustración 6.1 Logos de sistemas operativos famosos

6.1. Sistemas de uso personal

En este apartado se va a hablar de los sistemas operativos que sirven como base para los diferentes dispositivos que usan los usuarios, desde ordenadores a dispositivos portátiles como los móviles.

Es importante destacar que son estos sistemas los que más información del usuario recolectan, muchas veces para mejorar simplemente la experiencia de usuario, o quizá el rendimiento del dispositivo para un mejor consumo energético, aunque esta información puede tener otros usos diferentes una vez es recolectada, aunque el usuario no lo vea.

Un claro ejemplo puede ser la recolección de información sobre el uso que hace el usuario de diferentes aplicaciones del sistema operativo, como puede ser un explorador de archivos o simplemente la auditoria sobre las aplicaciones que un usuario instala y desinstala, o las tareas que se programan para su posterior ejecución.

Aunque el usuario vea como el tratamiento de esta información produce un comportamiento mejorado en el sistema (por ejemplo, en Windows 10 podemos acceder a través del explorador de archivos a accesos rápidos, que son ubicaciones o archivos que más se han usado) también puede servir en futuras actualizaciones del explorador de archivos del sistema operativo en como indexa esta información, o en cómo se presenta al usuario, basando estas decisiones en toda esa información recolectada [17].

6.1.1. Windows

Consultando la información disponible sobre la política de privacidad sobre el sistema operativo Windows, podemos encontrar tanto la información que se recopila, como se usan y el uso de que hacen de estos datos [18].

Como bien indican en su página, Microsoft recopila datos a través de las interacciones de los usuarios con sus sistemas, además de obtener datos de terceras aplicaciones. En cuanto al tipo de información que se recopila, podemos encontrar mucha variedad [1]:

- Nombre y datos de contacto
- Credenciales
- Datos demográficos
- Datos de pago
- Interacciones
 - Historiales de pagos, cuenta y de exploración
 - Datos de dispositivos, conectividad y configuración
 - Informes de errores y datos de rendimiento
 - Datos de solución de problemas y de ayuda
 - Datos de uso de bots
- Intereses y favoritos
- Datos de consumo de contenido
- Búsquedas y comandos
- Datos de voz
- Texto, entrada manuscrita y datos de escritura por teclado
- Imágenes
- Contactos y relaciones
- Datos sociales

- Datos de ubicación
- Contenido
 - Comunicaciones
 - Fotos, imágenes, canciones, películas, software y otros medios o documentos que se almacenen en su nube
- Comentarios y clasificaciones

Podemos ver que, prácticamente, Microsoft nos está diciendo que al hacer uso de su sistema operativo y usar parte del software que viene incluido en este, estamos aceptando compartir toda esa información con ellos.

En cuanto al propósito para la recopilación de toda esta información, y de manera resumida, Microsoft nos dice que recopila esta información para:

- Proporcionar nuestros productos, lo que incluye la actualización, protección y solución de problemas, así como facilitar soporte técnico.
- Mejorar y desarrollar nuestros productos.
- Personalizar nuestros productos y hacer recomendaciones.
- Anunciarle y comercializarle, lo que incluye el envío de comunicaciones de carácter promocional, publicidad dirigida y presentarle ofertas relevantes.

Indagando un poco más en que tratamientos y que propósito tiene, a parte de lo ya mencionado, en su página se menciona la investigación:

- Investigación. Con medidas técnicas y organizativas adecuadas para proteger los derechos y libertades de los individuos, usamos datos para realizar investigaciones, incluyendo fines científicos y de interés público.

Además de otros usos con fines propios de negocio, que a primera vista no se le cuentan al usuario. Cabe destacar que, aunque el usuario elija unas opciones de privacidad en el sistema, puede que no sea suficiente para evitar que el sistema operativo mande información a Microsoft [2].



Ilustración 6.2 Logo de Windows 10

6.1.2. Apple

Consultando la política de privacidad de la compañía Apple, que ofrece los sistemas operativos macOS en sus propios portátiles, e iOS en sus propios teléfonos móviles, podemos encontrar también una descripción de la información del usuario que es recolectada y el propósito de recolectarla [3].

De carácter personal, encontramos que se incluye información de recopilación de:

- Nombre
- Dirección postal

- Número de teléfono
- Dirección de correo electrónico
- Preferencias de contacto
- Identificadores de dispositivo
- Dirección IP
- Ubicación
- Datos bancarios

Esta información, de nuevo, es recopilada cuando el usuario interactúa con las diferentes aplicaciones que el sistema operativo incluye, o simplemente usando el mismo sistema operativo.

En cuanto a la justificación para la recolección de estos datos, podemos encontrar unas explicaciones un poco confusas pero que se parecen mucho a las dadas por Microsoft:

- Mantener al usuario informado sobre los últimos productos
- Para realizar mejores productos
- Evitar fraudes y la seguridad de los servicios que ofrecen
- Envío de notificaciones importantes

Por último, vemos que al igual que Microsoft y de manera muy breve, hacen referencia a investigaciones internas dando como excusa la mejora de sus productos:

- Uso de los datos con fines internos (como auditorias, análisis de datos e investigaciones)



Ilustración 6.3 Logo de iOS

6.1.3. Google

Consultando la política de privacidad de Google, que afecta también a sus distribuciones del sistema operativo Android, encontramos la información que esta compañía recoge y los usos para los que recolecta la información [4].

Para comenzar destacan que, estemos registrados o no con una de las cuentas gratuitas que ofrecen, vamos a estar identificados en todo momento a través de identificadores únicos, que pueden o no vincularse a una cuenta.

En cuanto a dicha información, no se explicita ninguna, pero si dejan claro que recolectan toda aquella información que pueda serles útiles, tanto para ofrecer una experiencia personalizada (es decir, contenido a la carta para el usuario, lo que suele ser recomendaciones y publicidad en base al uso del usuario) como para mejorar sus productos y ofrecer un servicio mejor.

Sobre el uso que hacen de la misma, también se menciona que esta información es procesada usando algoritmos y mecanismos automatizados, que además de ayudar en lo ya mencionado, también tiene un factor para detectar posibles usos fraudulentos o software malicioso.

Por último, también se menciona que pueden combinar la información que recogen mediante el software que tienen disponible y la información que recogen de los dispositivos que utilizan la distribución del sistema operativo Android de Google.



Ilustración 6.4 Logo de Android

6.1.4. Ubuntu

Como sistema operativo basado en el kernel⁶ Linux⁷ se ha elegido el sistema operativo Ubuntu, siendo este uno de los más famosos y más usados del mundo. En cuanto a la información que es recolectada, podemos encontrar muy poca [5]:

- Nombre completo
- Email
- Contraseña

Respecto a los medios de pagos, no se guardan, sino que simplemente se utilizan para procesar los pagos y nada más.

También se recolecta información sobre el uso que hace el usuario en el sistema operativo (muy poca, eso sí) y siempre se anonimiza, no identificando al usuario que hace uso del diferente software que incluye el sistema operativo.

En cuanto al uso que se hace de esta información, se menciona la mejora de los productos que ofrecen y estudiar (de manera anónima) como interactúan los usuarios con el software que proveen.



Ilustración 6.5 Logo de Ubuntu

⁶ [https://es.wikipedia.org/wiki/N%C3%BAcleo_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica))

⁷ <https://es.wikipedia.org/wiki/GNU/Linux>

6.2. Compañías de software

A continuación, se va a hablar de ciertas compañías muy importantes que proveen tanto software como servicios, y que durante los últimos años han estado en el punto de mira por sus políticas de privacidad y tácticas de negocio muy agresivas para captar usuarios.

Estas compañías, mediante diferente software, acaban recopilando grandes cantidades de información que tienen un gran valor ya que usan esta información para ofrecer servicios a terceras empresas, que no tienen ni los recursos ni la infraestructura para llegar a grandes cantidades de usuarios.

Es importante destacar que las dos compañías elegidas en este apartado usan la recolección de información de maneras diferentes, pero con un objetivo claro: obtener beneficios.

Mientras que Facebook utiliza la información recolectada para, mayormente, catalogar a sus usuarios y poder tener más efectividad en sus anuncios, Amazon utiliza su información para optimizar sus almacenes e intentar predecir el consumo de sus usuarios.

6.2.1. Facebook

Facebook es una de las compañías de software más grandes del mundo, siendo además una de las primeras redes sociales del mundo (tal y como se conocen hoy en día) y es famosa por sus múltiples escándalos en cuanto al uso y recolección de datos de sus usuarios se refiere.

Ya que es la mayor red social del mundo, se puede decir que la cantidad de información que tiene es muy grande, y mediante la adquisición de otras compañías de software [6] ha reforzado aún más su presencia en el día a día de las personas.

En cuanto a la privacidad de sus usuarios se refiere, Facebook recolecta la siguiente información [7]:

- Información y contenido que proporciona el usuario mediante el uso de sus servicios
- Redes y conexiones
- Interacciones del usuario con sus servicios
- Información sobre las transacciones realizadas con sus servicios
- La actividad de otros usuarios y la información que comparten del usuario
- Características de los dispositivos
- Operaciones del dispositivo
- Identificadores únicos del usuario
- Señales del dispositivo
- Datos de la configuración del dispositivo

Además, al proveer servicios a los usuarios y tener una comunidad de desarrolladores que expanden las funcionalidades de sus servicios, declara que estos también pueden compartir información con Facebook.

En cuanto a cómo utilizan esta información, encontramos lo siguiente:

- Para personalizar y mejorar sus productos
- Vincular la información entre dispositivos y servicios que ofrece Facebook
- Para mejorar y personalizar los anuncios del usuario
- Para investigación y desarrollo de productos
- Para identificar al usuario en diferentes contenidos

- Para proporcionar mediciones, análisis y otros servicios para empresas
- Para fomentar la seguridad, integridad y protección
- Para comunicarse con el usuario
- Para realizar investigaciones e innovar en beneficio el bienestar social



Ilustración 6.6 Logo de Facebook

6.2.2. Amazon

Amazon es una de las compañías más grandes del mundo, que empezó como un ecommerce y que poco a poco ha ido expandiendo los servicios que ofrece. Hoy en día recopila información de sus usuarios gracias a todos esos servicios que ofrece.

Aunque no ha estado completamente en el ojo del huracán por sus políticas de privacidad [19], sí que se ha demostrado el impacto que tiene en su negocio toda la información que colectan. Además, no solo toma decisiones estratégicas para el desarrollo de software, sino que basa también parte de sus decisiones de negocio en los resultados del análisis de esta información [8].

En cuanto a la información que recolecta Amazon de sus usuarios, podemos encontrar los siguientes medios [9]:

- Mediante la búsqueda de productos o servicios
- Realización de pedidos
- Publicación de una opinión
- Listas de deseos
- Descargas
- Streaming
- Visionado o uso de contenido digital en un dispositivo
- A través de los ajustes de la cuenta
- Configurando los ajustes de privacidad del dispositivo
- Hablando con Alexa⁸
- Comunicándose con la propia Amazon
- Nombre, dirección y número de teléfono
- Información de pago
- Datos de entregas
- Grabaciones de voz al hablar con Alexa

Sobre las finalidades de recabar esta información, se puede leer:

- Compra y entrega de productos y servicios
- Prestación, resolución de problemas y mejora de los servicios
- Recomendaciones y personalización
- Prestación de servicios de voz

⁸ <https://www.xataka.com/basics/que-alexa-que-puedes-hacer-que-dispositivos-compatibles>

- Cumplimiento de obligaciones legales
- Comunicarse con el usuario
- Publicidad
- Prevención del fraude y riesgos crediticios



Ilustración 6.7 Logo de Amazon

6.3. Compañías de hardware

A continuación, se hablará de varias compañías de hardware que hacen uso de la distribución de Android ofrecida por Google, sobre la que añaden una capa más de personalización, y, por lo tanto, software propio.

Estas compañías adquieren una posición ventajosa a la hora de ofrecer sus dispositivos ya que no solo añaden aplicaciones propias, sino que, mediante diferentes acuerdos, añaden software de terceras compañías.

Esto acaba teniendo un impacto importante en la privacidad del usuario, porque una vez empieza a usar el dispositivo, puede estar cediendo información a compañías de las que ni siquiera tiene conocimiento, aunque en un principio haya aceptado un mínimo de acuerdos de privacidad.

Además, aunque los terminales no deberían de tener ningún impedimento a la hora de desinstalar este tipo de aplicaciones que no son críticas para usar el dispositivo, el usuario se encuentra muchas veces con que no puede desinstalarlas, por lo que estas perduran durante toda la vida de uso del dispositivo, recibiendo incluso actualizaciones.

Por esto es que se ha decidido consultar sus políticas de privacidad y comprobar la cantidad de información que recolectan.

6.3.1. Samsung

Samsung es uno de los mayores distribuidores de dispositivos con sistema operativo basado en Android, siendo además uno de los que tiene una capa por encima de la ofrecida por Google más invasiva [10].

Al tener prácticamente una implementación propia para cada parte del sistema operativo Android (siempre por encima de lo ofrecido por Google, que es una implementación de ciertos componentes del teléfono) hace que básicamente recolecte la misma información que recolecta Google al hacer uso de sus servicios.

En su página podemos encontrar la siguiente información sobre la recolección de datos [11]:

- Toda la información del dispositivo
- Información de registro: información de diagnóstico, errores y uso
- Información de ubicación
- Información sobre redes
- Información de voz
- Información de teclado
- Información de uso

Además, se especifica que básicamente se recolecta cualquier otra información sobre el usuario del dispositivo no explicitada y sobre uso del dispositivo.

En cuanto al uso que se hace sobre esta información, podemos encontrar:

- Para registrar al usuario en diferentes servicios
- Para prestar servicios
- Para proporcionar contenido y servicios personalizados
- Para fines publicitarios

- Para análisis y evaluación: del mercado, los clientes, los productos y los servicios que se ofrecen
- Para solicitar opiniones
- Para entender como usan sus servicios los usuarios
- Para proporcionar servicios de mantenimiento
- Para facilitar acceso a actualizaciones
- Para cualquier otra finalidad

Además, explicitan que esta información es combinada, tanto de los dispositivos como de los servicios que ofrecen.

También se comparte cuando y con quienes se comparte toda esta información recolectada:

- Filiales
- Socios comerciales
- Prestadores de servicios
- Otras partes, cuando lo exija la ley o sea necesario para proteger sus servicios
- Otras partes en relación con transacciones corporativas
- Otras partes, con consentimiento o por petición del usuario



Ilustración 6.8 Logo de Samsung

6.3.2. Huawei

Huawei es otra de las grandes compañías de dispositivos móviles Android, que, aunque durante años ha estado basando sus distribuciones para estos dispositivos en la proporcionada por Google [10], ha sido vetada por el gobierno de los EEUU para el uso de esa distribución [12].

A pesar de este veto, todos los dispositivos vendidos antes del mismo pueden seguir usando estos servicios y recibir actualizaciones. Ante esto, Huawei ha empezado a distribuir teléfonos basados en su propia distribución de Android, lo que supone no poder usar la implementación base de Google, y aunque pierda ciertos servicios exclusivos, será Google quien deje de recolectar información a través de esos dispositivos (obviamente seguirá obteniendo información del uso de sus servicios mediante el software que descargue el usuario).

En cuanto a la información que Huawei recopila de sus usuarios, encontramos [13]:

- Información de aplicaciones y del dispositivo
- Información de la red móvil
- Información de registros
- Información de ubicación
- Información guardada en los servidores Huawei
- Información no identificable: estadísticas de visitas de sitios web o descargas de aplicaciones

Sobre los usos que hace Huawei de estos datos, vemos:

- Registrar y activar productos Huawei
- Enviar, activar o verificar productos solicitados
- Enviar actualizaciones del sistema y de las aplicaciones
- Proporcionar contenido y experiencia de usuario personalizada
- Enviar información sobre productos
- Llevar a cabo auditorías internas, análisis de datos e investigaciones
- Sincronizar, compartir y guardar datos que se hayan descargado
- Mejora de los programas de prevención de pérdidas y lucha contra el fraude
- Procesar datos de conformidad con las leyes y normas
- Otros fines de servicios específicos

También se menciona como Huawei comparte estos datos, con quien y cuando:

- Con terceros específicos cuando el usuario lo consienta
- En cumplimiento de una obligación legal
- Con proveedores de servicio de Huawei



Ilustración 6.9 Logo de Huawei

7. Conclusiones

En los capítulos anteriores se ha hecho un análisis sobre la privacidad de los usuarios en los diferentes sistemas de los que hacen uso en su día a día, explicando como la diferente información es extraída de una manera u otra.

Ya sea mediante el uso de dispositivos móviles como el uso de servicios a través de un ordenador, las compañías extraen toda la información que consideran que puede ser de utilidad o que, a la larga, les puede ayudar a tomar decisiones de negocio clave.

A lo largo de los años se ha podido ver el resultado de esta recolección masiva de datos, viendo como las compañías tecnológicas cada vez se han hecho más importantes y como poco a poco abarcan más sectores. No han estado fuera de polémica estos resultados, viendo como consecuencia como ciertas compañías se han tenido que enfrentar a juicios por la cesión de estos datos a terceras personas, o por la realización de prácticas poco éticas con sus usuarios.

Es aquí donde debe entrar la responsabilidad de los usuarios de valorar las diferentes opciones de privacidad que ofrecen las compañías, y aceptar que por el uso de sus servicios nuestra información va a ser recolectada, o por el contrario tener la opción de no aceptarla, con lo que conllevaría en muchos casos no disfrutar de un servicio o de un dispositivo.

Esta responsabilidad debería estar más arraigada en los usuarios, ya que muchas veces con la intención de disfrutar de un servicio, no se es consciente de lo que verdaderamente las compañías hacen con nuestros datos. Sea como fuere, si en algún momento un usuario decidiese dejar de usar un servicio fuese el que fuese, debería tener la posibilidad de poder deshabilitar este por completo.

Esto es una realidad en los ordenadores, en los que los usuarios tienen un control absoluto sobre las aplicaciones que están instaladas en los sistemas, y donde pueden poner barreras de por medio para cualquier recolección de información e incluso elegir qué servicios se ejecutan. Además, ya que muchos de los servicios que en los dispositivos móviles están presentes en forma de aplicaciones, en los ordenadores son accedidos mediante navegadores web, no suponen un mayor problema cuando un usuario simplemente quiere prescindir o dejar de usar cierto servicio. Incluso existen medios para limitar o bloquear por completo esta recolección de información, sin que el usuario pierda estos servicios.

En cambio, cuando hablamos de dispositivos móviles, los usuarios ven como están completamente atados a las aplicaciones que bien se proveen con el sistema operativo, o que las diferentes compañías de software. Muchas de estas aplicaciones, que ya están presentes con los dispositivos, no se pueden desinstalar, haciendo que, aunque no estemos formalmente identificados, tengamos un identificador único y que se recolecte nuestra información, ya que no podemos eliminarlas del dispositivo. Es cierto que para ciertos accesos de información sí que necesitan la aprobación por parte del usuario, pero no para el acceso a toda la información. Si estas aplicaciones no son necesarias para el desempeño correcto de las funcionalidades del dispositivo, ¿por qué tienen que estar presentes en el dispositivo cuando el usuario no lo desea?

No solo estas prácticas están muy limitadas en los dispositivos móviles actuales, sino que, además, una de las alternativas se basa en primero obtener acceso root⁹ en el dispositivo para poder desinstalar estas aplicaciones que vienen incluidas en los dispositivos, haciendo que al adquirir estos privilegios muchas funcionalidades del dispositivo queden deshabilitadas e incluso en algunos casos, que el usuario pierda la garantía del dispositivo.

⁹ https://es.wikipedia.org/wiki/Android_rooting

7.1. La privacidad en los dispositivos móviles Android

Cuando un usuario enciende y configura por primera vez un dispositivo, debe aceptar todas las condiciones que le vayan apareciendo, teniendo un mínimo de configuración, pero nunca teniendo la posibilidad de deshabilitar esta recolección de información por completo.

En cuanto a estas opciones de privacidad, en ninguna se hace referencia a la obligación de usar cierta aplicación o del derecho a desinstalar las aplicaciones o paquetes que el usuario considere, por lo que no hay una explicación clara a por qué no puede el usuario desinstalar lo que desee de su dispositivo.

Es por ello que debería ser lógico que los usuarios decidiesen qué aplicaciones (que no afecten al correcto funcionamiento del mismo) deberían estar presentes en sus dispositivos, teniendo en cuenta que como se ha mostrado, la mayoría de proveedores de servicios mediante estas aplicaciones recogen información del usuario.

Es aquí donde se encuentra sentido a una de las cláusulas de privacidad que suelen tener las compañías, y son los acuerdos para incluir entre ellas diferente software propietario para que la recolección de información sea en la mayor cantidad de usuarios posibles.

7.2. Medidas preventivas

Como se ha expuesto a lo largo de este documento en más de una ocasión, debería ser el usuario quien tuviese un control total de quién, cómo y cuándo recoge su información, no teniendo que estar atado a ningún servicio ni aplicación que viene proporcionada con el sistema operativo del dispositivo.

Ante estas prácticas, queda también claro que lo mejor para la privacidad del usuario siempre será tener en su dispositivo las aplicaciones que quiera utilizar y desinstalar todas aquellas que no vaya a usar, para al menos tener un mayor control sobre la información.

III. Fase 2: Desarrollo de la aplicación

8. Introducción

Tras lo expuesto en la fase anterior, podemos ver cómo dependiendo del sistema, podemos configurar o de alguna manera elegir qué se está ejecutando en nuestro sistema. Además, como ya se ha expuesto, esto no ocurre en los dispositivos móviles, en los que nos encontramos que ciertas aplicaciones, a veces, no pueden desinstalarse.

La parte de desarrollo de este trabajo tiene como base crear una aplicación que ayude a usuarios finales sin conocimientos avanzados a desinstalar cualquier paquete presente en el dispositivo. Esta parte del trabajo documenta cómo se ha llevado a cabo esta implementación, además de nombrar y explicar otras herramientas avanzadas ya existentes que incluyen esta funcionalidad.

Tras esta fase, se conseguirá una aplicación completamente funcional, con los requisitos deseados y que un usuario final puede utilizar.

9. Tecnologías y herramientas

En este capítulo se describen las herramientas y tecnologías que se han usado para el desarrollo de la aplicación, además de estudiar otras que, aunque no se han usado, son imprescindibles para entender de donde nace la necesidad de esta aplicación y como otras herramientas ya existentes pueden hacer lo mismo que esta aplicación usando otros medios.

9.1. Android

Android¹⁰ es un sistema operativo *Open Source*¹¹ sobre el que corren la mayoría de teléfonos móviles en el mundo. La mayoría de proveedores de estos dispositivos usan *Android Open Source Project*¹² o *AOSP* para crear imágenes propias, a las que añaden capas (tanto de sistema operativo como de interfaz de usuario) y generan un producto final, que se podría decirse es una distribución propia de Android.

Gracias a que estos sistemas estén basados en Android, y a que la aplicación desarrollada se basa en herramientas proporcionadas por Android, hace que la aplicación pueda usarse en cualquier dispositivo, sin importar siquiera la versión de Android ni el proveedor de la distribución que use el dispositivo.

9.1.1. Particiones

Como cualquier otro sistema operativo, Android tiene una distribución propia de particiones en las que almacena en cada una ellas partes del sistema para funcionar. Las principales son:

- **Boot:** tiene la imagen del kernel y los recursos de la RAM combinadas
- **System:** tiene el framework Android
- **Recovery:** tiene un pequeño sistema propio y ajeno al resto que permite realizar operaciones específicas y con privilegios de administrador
- **Cache:** tiene los datos temporales
- **Misc:** partición usada por la partición *recovery*
- **Userdata:** tiene las aplicaciones instaladas por el usuario y sus datos, además de las opciones de configuración del sistema
- **Metadata:** es la partición que se usa cuando el sistema está encriptado
- **Vendor:** contiene todos los binarios que no son distribuidos por AOSP
- **Radio:** software específico para el hardware propio de los teléfonos móviles como radio, GPS, bluetooth, WiFi, etc.
- **Tos:** tiene la imagen binaria de *Trusty OS*¹³ si es que el dispositivo incluye *Trusty*

Sobre estas particiones, en lo referente al trabajo que se quiere desarrollar y al objetivo que se quiere conseguir, *recovery* tiene una gran importancia, ya que es la que contiene los privilegios suficientes para realizar modificaciones en el resto.

¹⁰ <https://es.wikipedia.org/wiki/Android>

¹¹ https://es.wikipedia.org/wiki/C%C3%B3digo_abierto

¹² <https://source.android.com/>

¹³ <https://source.android.com/devices/bootloader/partitions/tos-partitions>

Aunque tenga los privilegios suficientes, normalmente estas acciones para hacer modificaciones no están disponibles a través de los *recovery* que incluyen los dispositivos, por lo que es necesario instalar uno modificado, llamado *custom recovery*, que si permita operaciones más avanzadas en el dispositivo.

9.1.2. Custom recovery

Como se acaba de mencionar, un *custom recovery*¹⁴ es una imagen *recovery* que no es la original, y que se instala en la partición *recovery*.

El único motivo por el que esta partición es tan interesante de modificar es que los *custom recovery*s extienden la funcionalidad de los originales. Esto nos permite hacer infinidad de acciones en el dispositivo, ya que no tenemos límites de acceso a cualquier parte del sistema:

- Backups
- Instalación de aplicaciones con privilegios root
- Modificación de otras particiones del sistema
 - Modificación de la partición *system* para cambiar la distribución de Android que usa el dispositivo
 - Modificación de la partición *boot* para cambiar el *kernel* que usa el dispositivo

En cuanto a lo relacionado a la aplicación desarrollada, para conseguir el mismo resultado, deberíamos de instalar una aplicación que tuviese privilegios root y que los proporcionase a otras aplicaciones. Un posible ejemplo de aplicación es Magisk¹⁵, aplicación que solo puede ser instalada desde un *custom recovery* ya que como se ha comentado, se instala con privilegios especiales.

9.1.3. Magisk

Magisk¹⁵ es una aplicación para el sistema operativo Android que se instala desde un *custom recovery* ya que necesita privilegios *root* para funcionar.

Esta aplicación lo único que hace es preguntar al usuario si quiere otorgar estos privilegios *root* a otras aplicaciones, por lo que, teniendo esta aplicación instalada en nuestro dispositivo, podemos realizar cualquier modificación del sistema cuando este ya está iniciado. Es importante saber a qué aplicaciones se les otorga estos privilegios, ya que una aplicación maliciosa podría causar graves daños en el dispositivo e incluso robar y recolectar toda la información que introduce el usuario tanto en las aplicaciones como en el propio sistema.

Por lo tanto, con un uso adecuado y teniendo esta aplicación, podemos mediante aplicaciones de gestión de otras aplicaciones desinstalarlas cuando el propio sistema Android no nos lo permite. Un ejemplo de aplicación de este tipo es *Titanium Backup*¹⁶ que nos permite una gestión muy avanzada de las aplicaciones, gracias a esos privilegios *root* que se le conceden.

¹⁴ <https://www.xatakandroid.com/roms-android/que-twrp-como-se-instala-sirve-este-custom-recovery-para-android>

¹⁵ <https://www.xatakandroid.com/programacion-android/que-magisk-sirve-como-se-instala>

¹⁶ <https://elandroidelibre.espanol.com/2011/10/tutorial-cmo-usar-correctamente-el-titanium-backup.html>

9.1.4. Android Debug Bridge

*Android Debug Bridge*¹⁷ o *ADB* es una herramienta de línea de comandos que permite la comunicación con el dispositivo. Es una de las librerías disponibles en las herramientas de la plataforma¹⁸ de Android. Es un programa cliente-servidor que incluye tres componentes [20]:

- Un cliente que envía comandos
- Un *daemon*¹⁹ que ejecuta comandos en el dispositivo
- Un servidor, que administra la comunicación entre el cliente y el *daemon*

Gracias a esta herramienta vamos a poder conectarnos al dispositivo cuando esté conectado al ordenador mediante un cable USB. Para esto, es imprescindible que primero el usuario active el *modo debug* en el dispositivo, para que después de habilitar este modo y realizar la conexión entre el ordenador y el dispositivo, marque al ordenador como un dispositivo de confianza.

9.1.5. Android Shell

Mediante la herramienta *ADB* podemos usar el comando *shell* para iniciar un terminal interactivo dentro del dispositivo. Y gracias a este terminal interactivo vamos a poder usar ciertas herramientas que son habituales en entornos UNIX.

Para la aplicación que vamos a desarrollar, vamos a usar la misma herramienta para obtener información de las aplicaciones del sistema (ubicación, nombre de aplicación y nombre del paquete) como para desinstalarlos, y no es otra que el administrador de paquetes de Android, con diferentes comandos.

9.1.6. Android Package Manager

El gestor de paquetes de Android (*Android Package Manager*²⁰) es la herramienta dentro del terminal interactivo que vamos a usar. Mediante “*pm*” vamos a invocarlo, siendo los siguientes comandos los más útiles para la aplicación que queremos desarrollar:

- *pm list packages*: devuelve un listado de todos los paquetes
- *pm uninstall <package-name>*: eliminar un paquete del sistema

¹⁷ <https://developer.android.com/studio/command-line/adb?hl=es-419>

¹⁸ <https://developer.android.com/studio/releases/platform-tools>

¹⁹ [https://es.wikipedia.org/wiki/Daemon_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Daemon_(inform%C3%A1tica))

²⁰ <https://developer.android.com/studio/command-line/adb?hl=es-419>

9.2. Python

Python²¹ es un lenguaje de programación interpretado, multiparadigma que es muy sencillo de usar. Se ha decidido optar por este lenguaje para la implementación de la aplicación ya que muchos sistemas basados en UNIX²² ya lo incorporan por defecto.

En caso de que el sistema no tenga Python instalado, la instalación en cualquier entorno es muy sencilla ya que no requiere ningún tipo de configuración previa ni posterior, por lo que no sería un problema para un usuario que sepa simplemente instalar un programa.

Además, gracias a que Python, mediante el uso de ciertas librerías permite crear aplicaciones ejecutables que incorporan todo lo necesario para su ejecución, no haría falta tampoco la instalación previa de Python, haciendo que los usuarios finales simplemente tengan que ejecutar la versión ya creada para su sistema operativo.

²¹ <https://es.wikipedia.org/wiki/Python>

²² <https://es.wikipedia.org/wiki/Unix>

9.3. Git

Git²³ es una herramienta para el control de versiones Open Source que permite el desarrollo de un trabajo de manera iterativa e independiente entre los procesos que se puede descomponer una aplicación.

Para cada tarea a realizar se pueden usar diferentes *ramas*, que a su vez están compuestas por *commits*, que no son más que los cambios que queremos guardar. Estas ramas, cuando tienen todos los *commits* con los cambios que queremos añadir, se añaden a la *rama* principal mediante un *merge*, que significa incorporar a la *rama* principal los *commits* de la rama que hemos usado para una tarea.

Esta estrategia de trabajo puede variar, ya que podemos no trabajar con diferentes ramas y tan solo trabajar añadiendo los *commits* en la *rama* principal. En cualquier caso, lo que nos importa es cada cambio que marquemos mediante *commits* se vea reflejado.

²³ <https://es.wikipedia.org/wiki/Git>

10. Desarrollo de la aplicación

Después de haber explicado las herramientas disponibles para usar y desarrollar la aplicación, vamos a explicar en este capítulo como se ha diseñado la aplicación, tanto a nivel de arquitectura como de estructura de proyecto.

Se ha tenido en mente a la hora de desarrollar la aplicación la capacidad de iterar o de poder anexionar esta aplicación como una funcionalidad para otras aplicaciones. Teniendo esto como base, se ha intentado que el código esté lo menos acoplado entre sí, dando pie a que las modificaciones que puedan surgir afecten lo mínimo posible.

En lo que respecta a la usabilidad, se ha intentado hacer la aplicación lo más amigable y fácil de usar para que los usuarios no necesiten prácticamente a usar el producto, y que con la mera intuición y legibilidad de la aplicación sea casi suficiente para saber qué podemos hacer.

10.1. Arquitectura

Como arquitectura para la aplicación se ha optado por la arquitectura hexagonal [14], también conocida como arquitectura de puertos y adaptadores.

Esta arquitectura tiene como principal motivación separar las diferentes capas en las que se descompone una aplicación por dentro, teniendo cada una de ellas su propia responsabilidad. Esta separación nos permite además tener diferentes implementaciones en cada una de las capas, lo que otorga una gran flexibilidad a la hora de cambiar las propias implementaciones.

Por otro lado, se ha programado la aplicación teniendo en mente que pueda ser un componente de otra aplicación que quiera incluir esta funcionalidad, por lo que la puerta de entrada de datos es completamente accesible.

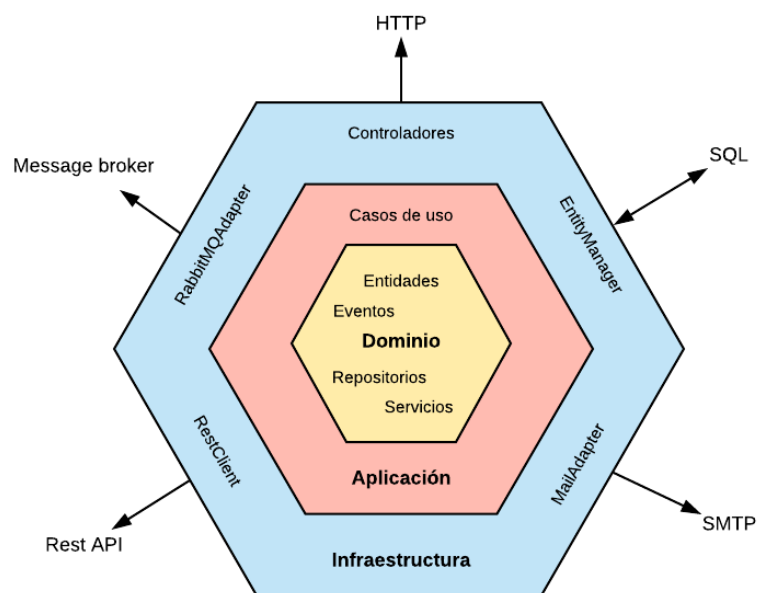


Ilustración 10.1 Representación de la arquitectura hexagonal

10.2. Conexión con el dispositivo

Para poder conectar la aplicación con el dispositivo, antes hay que realizar varias acciones en el mismo, que van a permitir enseñar al usuario unas opciones de configuración más avanzadas. Una de estas configuraciones es la de permitir el modo *debug*²⁴, que hace posible la conexión entre el dispositivo y las librerías que proporciona Google, conocidas como las herramientas de la plataforma o *platform tools*.

Para el caso en concreto de la aplicación, como ya se ha mencionado, se va a hacer uso de *ADB*, que requiere también de que este modo este habilitado. Para empezar, primero debemos habilitar estas opciones avanzadas, que en los dispositivos se llaman “Opciones de desarrollo”. Para ello, deberemos ir a los ajustes del dispositivo, a la información de la versión de compilación del sistema, y pulsar varias veces hasta que nos salga un banner advirtiéndonos de que estas opciones están habilitadas:

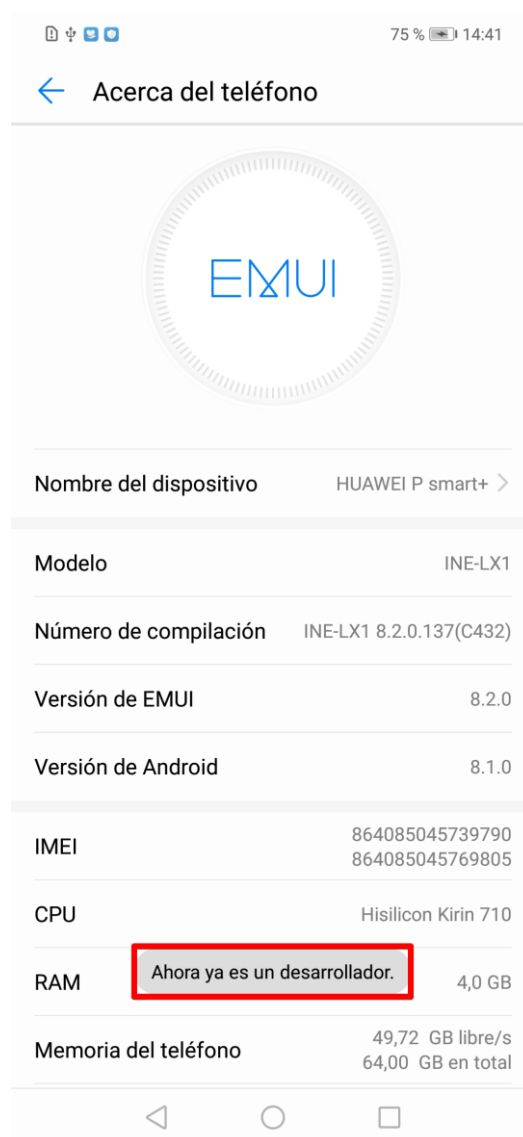


Ilustración 10.2 Activación de las opciones de desarrollo

²⁴ <https://developer.android.com/studio/debug/dev-options?hl=es-419>

Una vez estas opciones están disponibles, deberemos entrar en ellas y habilitar la opción “Depuración USB” que se refiere a la activación del modo *debug* en el dispositivo:



Ilustración 10.3 Activación del modo debug

Por último, una vez iniciemos la aplicación e intentemos cargar los paquetes del dispositivo, la primera vez nos saldrá en el dispositivo una pequeña ventana, preguntando al usuario si verdaderamente quiere confiar en el ordenador que está intentando conectarse al dispositivo. Una vez aceptemos, podremos ver todos los paquetes presentes en el dispositivo:

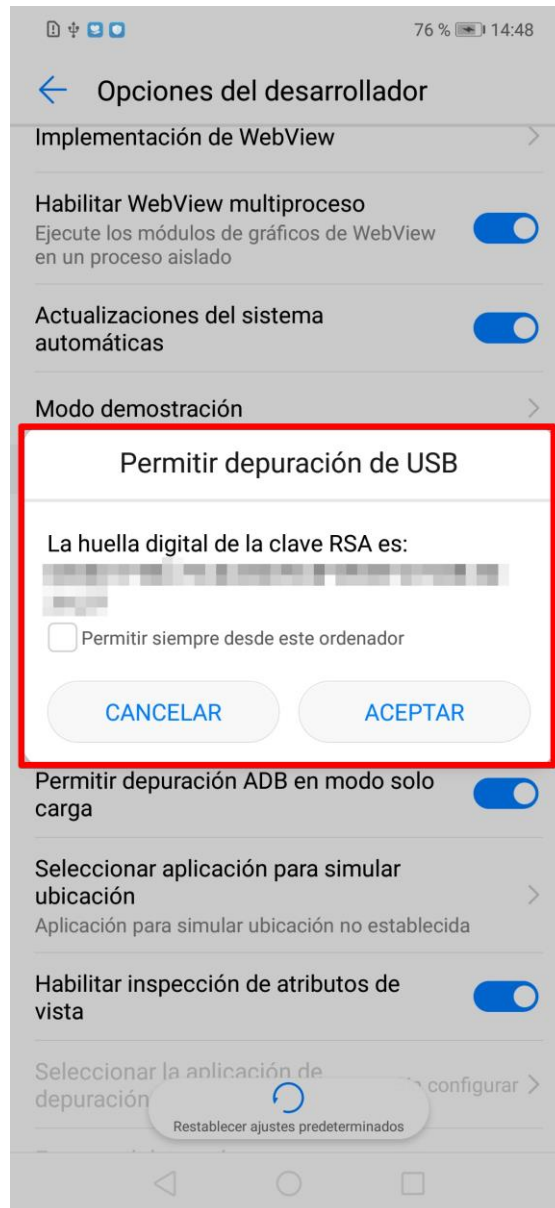


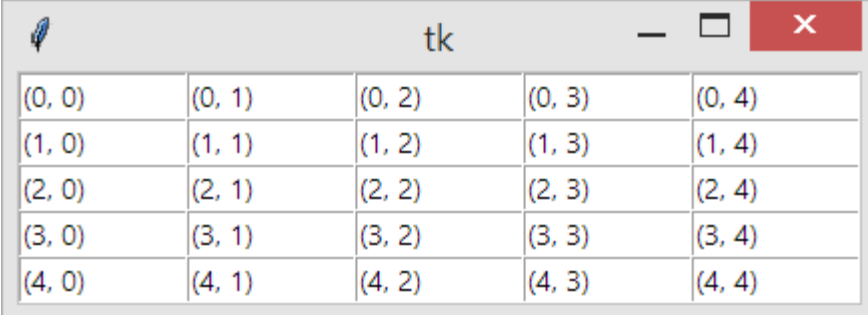
Ilustración 10.4 Ventana para permitir acceso del ordenador al dispositivo

10.3. Interfaz grafica

Para la interfaz gráfica se ha optado por una única ventana que permita hacer todas las acciones de la aplicación:

- Leer los paquetes del dispositivo
- Filtrar los paquetes del dispositivo dado un texto
- Modificar los paquetes del dispositivo

Para esta ventana única se ha usado una librería propia de Python llamada Tkinter²⁵ que, al estar integrada en el propio entorno de desarrollo de Python, no ha requerido instalación previa. Además, todos los componentes, su diseño y posición de la aplicación se han definido mediante líneas de código, lo que ha evitado tener que crear un primer diseño para posteriormente, mediante código, unir estos componentes con ese diseño.



(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)
(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)
(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)
(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)
(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)

Ilustración 10.5 Ejemplo de distribución de filas y columnas en Tkinter

²⁵ <https://wiki.python.org/moin/TkInter>

10.4. Navegación

Se ha querido que la navegación dentro de la aplicación sea lo más sencilla posible, dejando a la vista del usuario desde el primer momento las opciones disponibles. Es por esto que la ventana principal tiene a la vista desde el primer momento las acciones que están disponibles en la aplicación.

A parte de esto, toda la información que se obtiene del dispositivo es presentada en la misma ventana, por lo que no requiere que el usuario cambie de ventana o busque el resultado de las acciones que realiza.

Esta decisión viene motivada por el hecho de que se quiere un programa lo más simple posible, y que funcionalmente cumpla su objetivo sin que se necesite un aprendizaje de uso para la aplicación.

10.4.1. Listado de paquetes

Para que el usuario dentro de la aplicación pueda ver todos los paquetes instalados en el dispositivo, se ha añadido un botón específico que realiza la conexión con el dispositivo y presenta después todos los paquetes del mismo.

La conexión se hará mediante el paquete ADB y el entorno sobre el que se esté ejecutando la aplicación, ya que se lanzará según se pulse el botón el comando para iniciar el servidor de conexión y también la petición de obtención de la información de los paquetes.

Obviamente esta acción debe hacerse siempre con el dispositivo conectado, y no requiere de más interacciones por parte del usuario. Una vez obtenidas todos los paquetes, estos se muestran directamente en la ventana, sin ningún tipo de filtro y con el recuadro para filtrar vacío.

Una vez se aplica el filtro, solo se mostrarán las aplicaciones que contengan en el texto de cualquiera de las tres columnas de información disponibles el texto introducido.

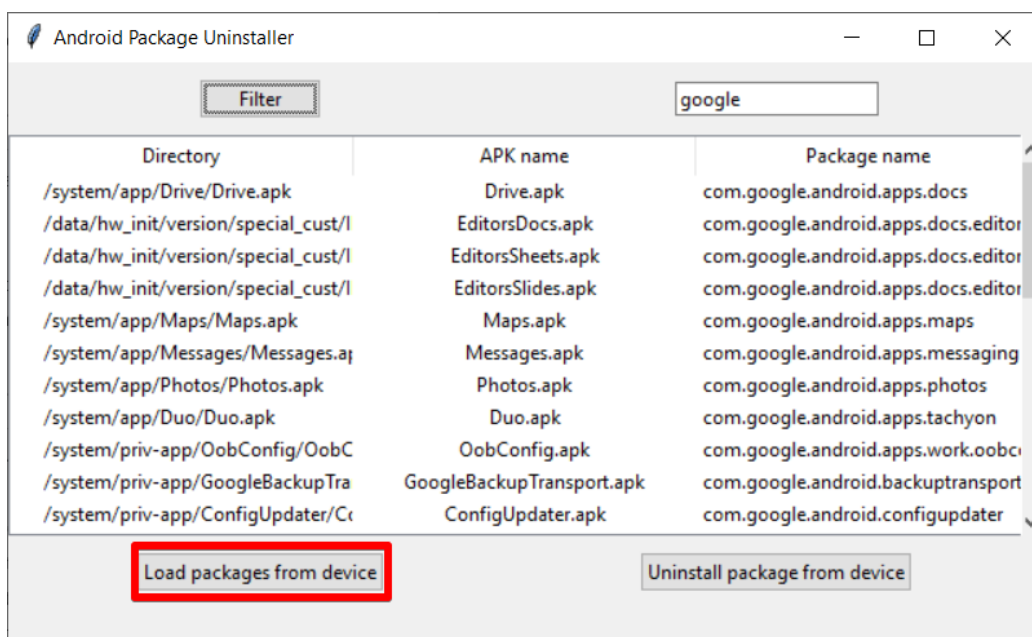


Ilustración 10.6 Botón de listar paquetes de la aplicación

10.4.2. Desinstalación de paquetes

Para el caso de la desinstalación de paquetes por parte del usuario, lo primero que se debe realizar es el listado de paquetes, ya que de otra manera no será posible elegir un paquete para su desinstalación.

Una vez se tiene el listado, vale con elegir haciendo clic en el paquete que se quiere desinstalar, quedándose este marcado de color azul para resaltar la opción elegida. Una vez elegida, bastará con elegir la opción de desinstalar, que lanzará el comando para la desinstalación del paquete.

Ya que previamente se habrá creado una conexión con el dispositivo para listar las aplicaciones, este comando actúa de manera casi inmediata.

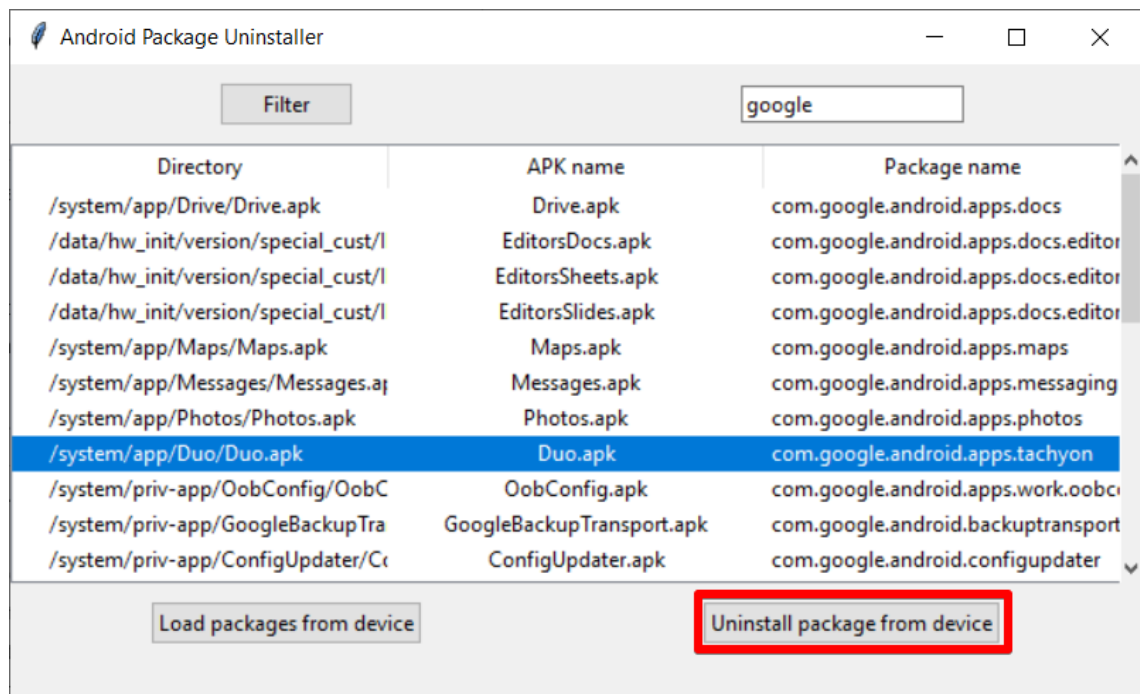


Ilustración 10.7 Botón de desinstalar paquetes de la aplicación

10.5. Conclusiones

Se han analizado las herramientas disponibles hoy en día para conseguir el mismo objetivo que se ha planteado para la aplicación desarrollada, viendo que se requieren muchos conocimientos técnicos, tanto de sistemas operativos (en este caso, de Android) como de la gestión específica de Android en cuanto a las particiones, para poder conseguir los recursos necesarios para desinstalar una simple aplicación.

Esto supone un problema de inicio para todos los usuarios que no tengan conocimientos técnicos o para los que no sepan cómo funcionan los sistemas basados en Android, ya que la falta de conocimientos puede acabar con un dispositivo completamente imposible de usar e incluso imposible de recuperar. Como las compañías son contrarias a este tipo de modificaciones en sus dispositivos, cuando un usuario intenta usar la garantía para arreglar un dispositivo móvil con el sistema corrompido (conocido coloquialmente en internet como *bricked phone*²⁶) se encuentra con la negativa de las compañías, que basándose en sus políticas de uso en las que no se permite este tipo de modificaciones, no hacen nada para ayudar al usuario a recuperar su dispositivo móvil.

Es por este motivo por el que hoy en día cada vez menos usuarios optan por iniciar este proceso que requiere la instalación de un *custom recovery*, para posteriormente instalar una aplicación con acceso *root* que además proporcione estos privilegios a terceras aplicaciones, y por último tener que instalar otra aplicación que sea compatible con los privilegios *root* para que haga uso de ellos y pueda desinstalar cualquier tipo de paquete en su dispositivo.

Como se ve, es un proceso engorroso que no está exento de cierto peligro, y es por este miedo que finalmente los usuarios optan por simplemente ceder ante esto y tener esos paquetes presentes en sus dispositivos.

En lo que al desarrollo de la aplicación de refiere, la interfaz gráfica y su conexión con la aplicación no ha sido muy difícil ni muy problemática, ya que esta se ha realizado completamente programándola mediante líneas de código. Esto ha hecho que la conexión entre eventos o entradas del usuario mediante los diferentes componentes utilizados haya resultado de lo más sencilla, que es lo que se buscaba a la hora de desarrollar la aplicación.

Aunque el desarrollo de esta interfaz haya resultado finalmente algo sencillo, lo que se ha querido buscar en todo momento es que el usuario vea de la manera más clara posible las acciones disponibles en la aplicación y que su uso sea lo más simple posible. De esta manera se consigue que la curva de aprendizaje sea muy baja o prácticamente nula, ya que los propios botones mostrados definen brevemente la acción que realizan.

Sobre la conexión entre la aplicación y el dispositivo, gracias al uso de las librerías de Python que abstraen a la aplicación sobre el sistema sobre el que se ejecuta, no ha requerido más cambios que definir comandos propios para cada sistema.

²⁶ <https://www.howtogeek.com/126665/htg-explains-what-does-bricking-a-device-mean/>

11. Implementación de la aplicación

En este apartado hablaremos de la implementación de la aplicación, teniendo como base todo lo mencionado anteriormente para conseguir una estructura y legibilidad del código lo más optima posible.

Esta parte es muy importante porque además de asentar las bases y el camino a seguir, tiene que ser lo más flexible posible para los futuros cambios que se puedan presentar en la aplicación, sin dejar de ser robusta.

Cabe destacar que esta sección no pretende ser una documentación del código, sino que es más una explicación y guía para entender mejor cómo se ha procedido a la hora de implementar las diferentes partes del proyecto.

11.1. Estructura del proyecto

El proyecto tiene un módulo principal que sirve para agrupar a su vez todos los módulos de los que se compone la aplicación. Esto es así porque al usar tantos módulos y necesitar exportar e importar las clases, Python no entiende de rutas relativas entre clases, por lo que agrupándolos todos dentro de un mismo modulo (el principal), sí que puede entenderse entre los módulos que hay dentro.

Una vez dentro de este módulo principal, encontramos otros cuatro módulos principales:

- **Application:** este módulo contiene los casos de uso de la aplicación. Además, contiene una abstracción de la capa de acceso de datos, que será quien conecte este módulo con el de */infraestructure/adapter*.
- **Domain:** este módulo contiene los objetos que representan lógica de negocio. En nuestro caso, solo tendremos los ya mencionados paquetes de Android.
- **Infraestructure:** este módulo contiene otros tres módulos. Esto es así porque esta capa representa la capa exterior de la aplicación, es decir, que recoge las acciones del usuario (puerto) y del dispositivo (adaptador). Además, tiene un tercer modulo que es la tarea que prepara el entorno con los archivos necesarios para funcionar.
- **View:** este módulo contiene la implementación de las vistas. En nuestro caso, al ser una única ventana, contiene un único archivo.

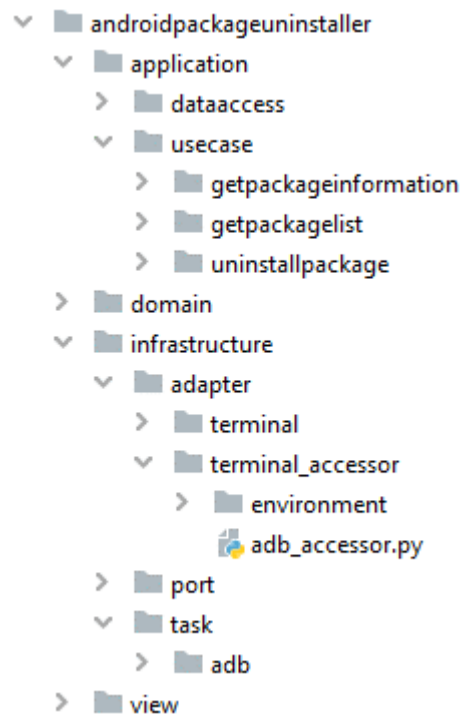


Ilustración 11.1 Estructura del módulo principal del proyecto

Además, en el directorio principal también tenemos otros archivos y directorios que funcionan junto con la aplicación. Lo más destacable aquí es:

- En el directorio `/adb` están los archivos descargados y descomprimidos para que la aplicación funcione
- En el directorio `/venv` tenemos el entorno virtual creado que contiene las librerías necesarias para que la aplicación funcione

Estos dos directorios se generan dinámicamente con el uso de la aplicación: `/adb` se descarga al lanzar la aplicación, y `/venv` es la preparación del entorno para trabajar sobre un sandbox²⁷ en vez del entorno del propio ordenador.

²⁷ [https://es.wikipedia.org/wiki/Entorno_de_pruebas_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Entorno_de_pruebas_(inform%C3%A1tica))

Por último, se incluye el archivo *.gitignore* que especifica los archivos a ignorar por git, un *README.md* con explicaciones de cómo preparar el entorno de desarrollo y lanzar la aplicación, un *requirements.txt* que tiene especificadas las librerías usadas junto con las versiones específicas, y el archivo *main.py* que se usa para lanzar la aplicación.

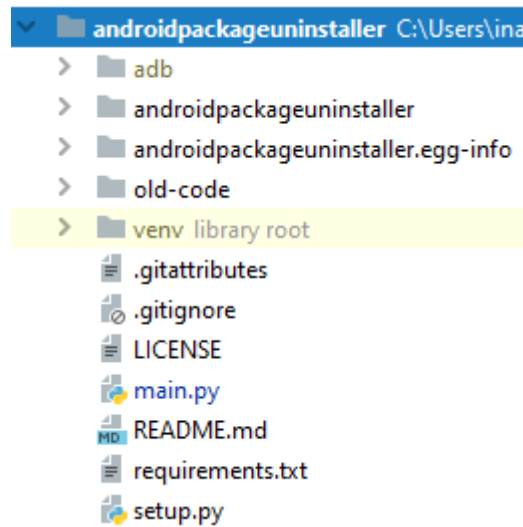


Ilustración 11.2 Estructura general del proyecto

11.2. Integración de ADB

Para que la aplicación pueda funcionar, necesitamos usar los paquetes presentes en */adb* que no son más que el propio Android Debug Bridge junto con otros paquetes. El uso de estos paquetes es mediante una línea de comandos o terminal, funcionalidad que nos facilita Python mediante una de sus librerías ya incluida.

Para poder usar ADB vamos a necesitar una tarea que consiga descargar los archivos necesarios, que son diferentes dependiendo del entorno desde el que se lance la aplicación, y por ultimo las interacciones con estos paquetes que, aunque Python nos ponga ciertas facilidades, varía entre los diferentes entornos.

11.2.1. Descarga

Lo primero que necesitamos es identificar el entorno desde el que se está lanzando la aplicación. De nuevo, Python nos provee de medios para identificar el entorno, por lo que una vez identificado, directamente descargamos la versión para ese entorno.

```
def __get_download_url(self, environment: str) -> str:
    # Obtenemos la URL de descarga dependiendo del entorno
    if environment == "linux" or environment == "linux2":
        return self.LINUX_DOWNLOAD_URL
    elif environment == "darwin":
        return self.DARWIN_DOWNLOAD_URL
    elif environment == "win32":
        return self.WINDOWS_DOWNLOAD_URL
    else:
        raise ValueError('Provided environment doesnt have any
download URL.')
```

Código fuente 11.1 Función de descarga de ADB

Una vez descargado, se procede a la descompresión del archivo descargado, dando como resultado un nuevo directorio con todos los paquetes necesarios para funcionar. Cuando se termina de descomprimir, el archivo descargado es eliminado.

```
@staticmethod
def __unzip_downloaded_file(file_name: str, extract_directory: str):
    # Descomprimos el archivo descargado
    file = ZipFile(file_name)
    file.extractall(extract_directory)
    file.close()
```

Código fuente 11.2 Función de descompresión de la descarga de ADB

```
@staticmethod
def __create_download_directory(directory: str):
    # Creamos la carpeta donde descargar los archivos
    os.makedirs(directory)
```

Código fuente 11.3 Función de creación de directorio para archivos descargados

```
@staticmethod
def __remove_downloaded_file(file_directory: str):
    # Eliminamos el archivo descargado
    os.remove(file_directory)
```

Código fuente 11.4 Función para eliminar archivo ADB descargado

11.2.2. Interacciones

Para poder usar los paquetes descargados, vamos a tener que usar de nuevo una librería incluida con Python, que nos facilitará invocar una terminal y que es independiente del entorno sobre el que este corriendo la aplicación.

Para cada uno de los entornos disponibles vamos a tener una implementación propia para lanzar los comandos para ADB, ya que, aunque Python nos abstraee a la hora de invocar la terminal, la manera de generar los comandos es diferente:

```
# Windows
ADB_DIRECTORY: Final[str] = ".\\adb\\platform-tools\\"
UNINSTALL_PACKAGE: Final[str] = "adb shell pm uninstall --user 0 "
LIST_ALL_PACKAGES: Final[str] = "adb.exe shell pm list packages -f"
```

Código fuente 11.5 Comandos para entornos Windows

```
# Linux
ADB_DIRECTORY: Final[str] = "./adb/platform-tools/"
UNINSTALL_PACKAGE: Final[str] = "adb shell pm uninstall --user 0 "
LIST_ALL_PACKAGES: Final[str] = "adb shell pm list packages -f"
```

Código fuente 11.6 Comandos para entornos Linux

```
# Darwin
ADB_DIRECTORY: Final[str] = "./adb/platform-tools/"
UNINSTALL_PACKAGE: Final[str] = "adb shell pm uninstall --user 0 "
LIST_ALL_PACKAGES: Final[str] = "adb shell pm list packages -f"
```

Código fuente 11.7 Comandos para entornos Darwin

Aunque los comandos para los entornos de Linux y Darwin son los mismos, se ha decidido dejarlos separados por si en un futuro esto cambia.

11.3. Uso de ADB

Para la ejecución de los comandos de ADB se ha usado la librería *subprocess* de Python. Gracias a esta librería y a la función *check_output* conseguimos ejecutar el comando deseado y obtener el resultado del mismo muy fácilmente.

Todos los comandos tienen en común una parte, y es la de crear una conexión a través de la terminal con el dispositivo. Esta conexión se hace a través de *adb shell* que de un modo normal y sin ningún otro parámetro, abriría una terminal dentro del dispositivo. Lanzando así el comando, indicamos que los siguientes parámetros son como si ya tuviésemos la terminal en el dispositivo abierta, lo que nos permite ejecutar otros comandos, como los que vamos a usar para el listado de paquetes con toda su información y para desinstalar cualquier paquete.

11.3.1. Lectura de paquetes

En el caso de la lectura de paquetes, ya que no necesitamos añadir ningún atributo al comando que queremos lanzar en consola, directamente formamos el comando con la localización del paquete ADB y el del comando de listado de paquetes.

En este caso, simplemente lanzaremos el comando *pm list packages -f* que nos mostrara todos los paquetes que tiene el dispositivo instalados, dándonos gracias al parámetro *-f* la ubicación del archivo instalado, el id del paquete y el nombre de la apk²⁸ que se instala:

```
def get_package_list(self) -> List[str]:
    # Preparamos el comando a lanzar en la terminal
    shell_command: str = self.ADB_DIRECTORY + self.LIST_ALL_PACKAGES
    # Lanzamos el comando y capturamos el valor, formateándolo
    output: List[str] =
        subprocess.check_output(shell_command).decode('utf-8')
        .split('package:')
    return output
```

Código fuente 11.8 Función para listar paquetes del dispositivo

11.3.2. Modificación de paquetes

En el caso de desinstalación de un paquete, necesitamos recibir el nombre del mismo para poder añadirlo al comando que se lanzará. Esto no es un gran inconveniente ya que simplemente añadiendo el valor al comando formado por la localización del paquete ADB y de desinstalación de paquetes, conseguimos lanzarlo correctamente.

Para este caso sí que debemos especificar el nombre del paquete que vamos a instalar, pero no supone más problema que añadirlo al comando a lanzar. Para desinstalar un paquete usaremos el comando *pm uninstall -user 0 <paquete>* que desinstalara para el usuario principal del dispositivo el paquete que se pase por parámetro:

²⁸ [https://es.wikipedia.org/wiki/APK_\(formato\)](https://es.wikipedia.org/wiki/APK_(formato))


```
def uninstall_package(self, package_name: str):  
    # Preparamos el comando a lanzar en la terminal  
    shell_command = self.ADB_DIRECTORY + str(self.UNINSTALL_PACKAGE)  
+ str(package_name)  
    # Lanzamos el comando y capturamos el valor, formateándolo  
    output: str = subprocess.check_output(shell_command)  
    .decode('utf-8')  
    return output
```

Código fuente 11.9 Función para desinstalar un paquete del dispositivo

11.4. Interfaz gráfica

Para la interfaz gráfica se ha usado la librería Tkinter, ya que esta se puede diseñar programáticamente. La ventana de la aplicación consta de tres botones y un cuadro para entrada de texto, que se usará para filtrar las aplicaciones mostradas.

Para agrupar las funcionalidades por secciones, se ha puesto en una primera sección el botón de filtrar junto con el cuadro para la entrada de texto, para que sea más entendible su uso.

```
# Cuadro de texto para filtrar
self.filter_value = Entry(main_window)
self.filter_value.grid(row=0, column=1)

# Boton para aplicar el filtro
filter_button = Button(main_window, text='Filter',
command=self.__filter_packages_by_name)
filter_button.grid(row=0, column=0)
```

Código fuente 11.10 Código para añadir entrada de texto y botón de filtrar

Mas abajo tenemos la segunda sección, que es para el listado de aplicaciones, en el que se pueden apreciar tres columnas: la primera columna muestra la ubicación del paquete instalado, la segunda columna muestra el nombre del paquete instalado y, por último, en la tercera columna, se muestra el id del paquete instalado.

```
# Creamos el componente para listar los paquetes
self.tree_view = Treeview(main_window, columns=('apk',
'package'))
# Creamos un scrollbar lateral
vsb = Scrollbar(self.tree_view, orient="vertical",
command=self.tree_view.yview)
# Posición en la que queremos fijar el scrollbar y tamaño
vsb.place(relx=0.980, rely=0.0, relheight=1.0, relwidth=0.020)
self.tree_view.configure(yscrollcommand=vsb.set)
# Primera columna
self.tree_view.heading('#0', text='Directory')
# Segunda columna, con alineado al centro
self.tree_view.heading('#1', text='APK name')
self.tree_view.column('#1', anchor=CENTER)
# Tercera columna
self.tree_view.heading('#2', text='Package name')
# Bindeamos la selección con una variable
self.tree_view.bind("<<TreeviewSelect>>", self.__on_tree_select)
self.tree_view.grid(row=1, column=0, columnspan=2,
sticky=N+S+E+W)
```

Código fuente 11.11 Código para añadir columnas con diferente información de los paquetes

En la última sección tenemos dos botones, el primero para obtener y mostrar la lista de todos los paquetes instalados en el dispositivo, y el segundo para desinstalar una de las aplicaciones que se haya elegido del listado mostrado.

```
# Botón para listar los paquetes
list_packages = Button(main_window, text='Load packages from
device', command=partial(self.__get_device_packages))
list_packages.grid(row=2, column=0)

# Botón para desinstalar el paquete seleccionado
uninstall_button = Button(main_window, text='Uninstall package
from device',

command=partial(self.__uninstall_device_package))
```

Código fuente 11.12 Código para añadir botón de listado de paquetes y desinstalación de un paquete

La ventana una vez lanzamos la aplicación, con todos los componentes correctamente cargados y sin mostrar información ya que aún no se ha interactuado con ella:

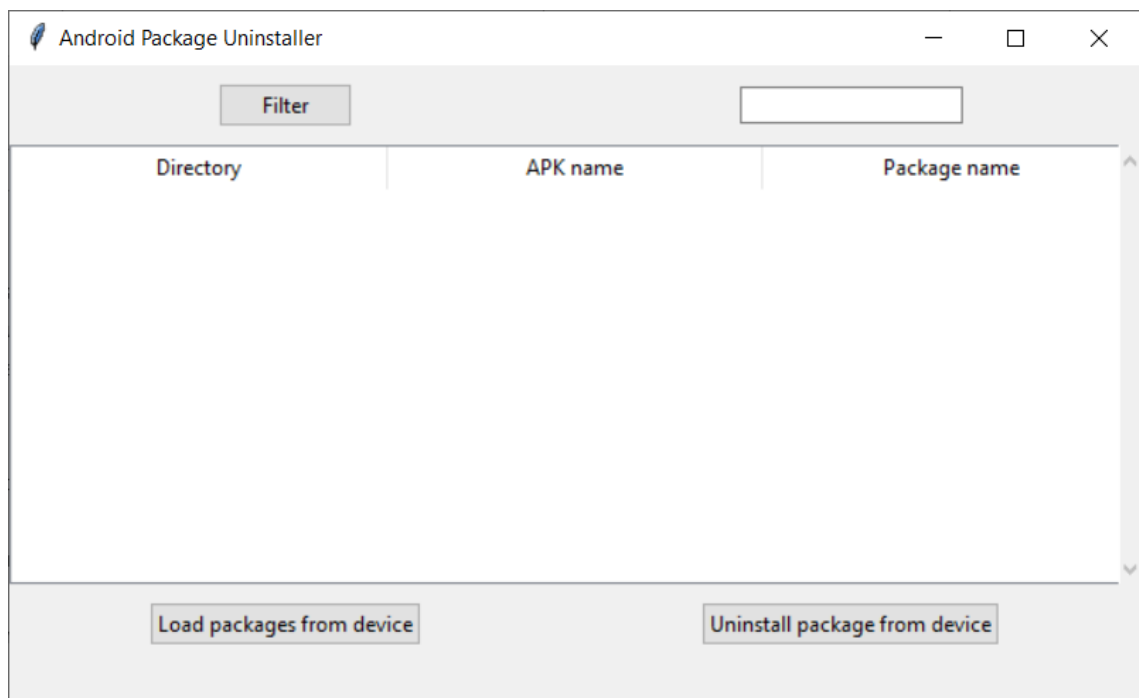


Ilustración 11.3 Ventana principal de la aplicación al ejecutarla

Aquí podemos ver la ventana una vez hemos pulsado el botón de cargar todas las aplicaciones del dispositivo, después de cargar todos los paquetes sin que haya ningún filtro aplicado:

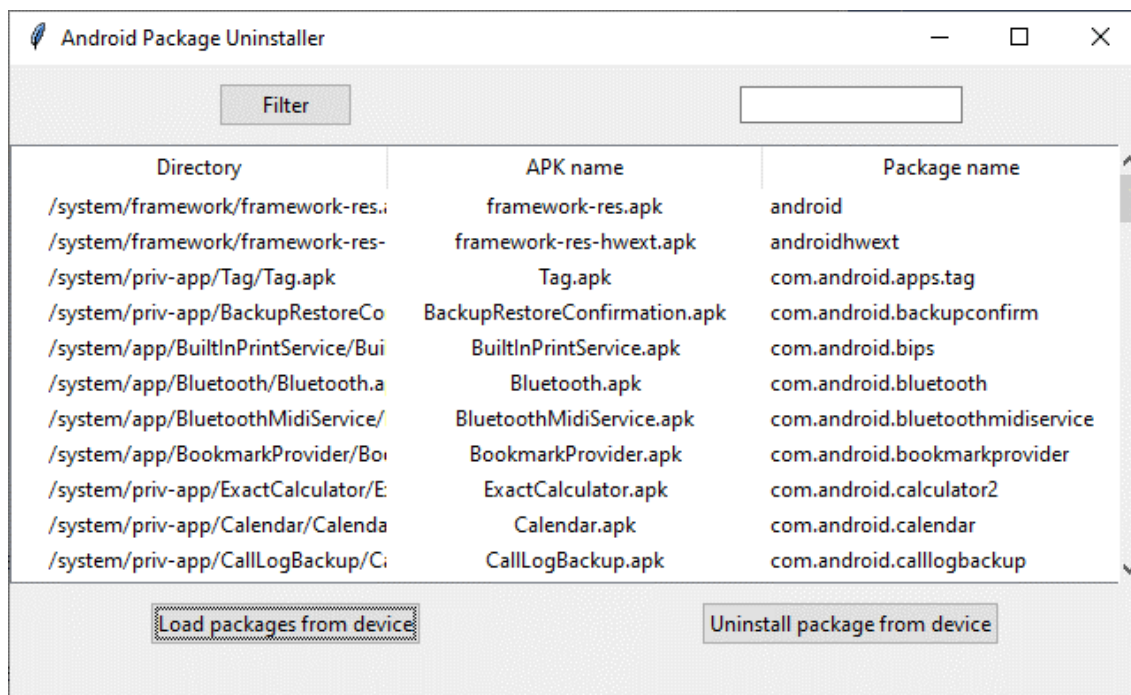


Ilustración 11.4 Ventana de la aplicación una vez listados los paquetes

Y por último la ventana una vez se usan tanto el cuadro para filtrar el texto como el botón para filtrarlo, enseñando tan solo los paquetes que incluyen el texto introducido:

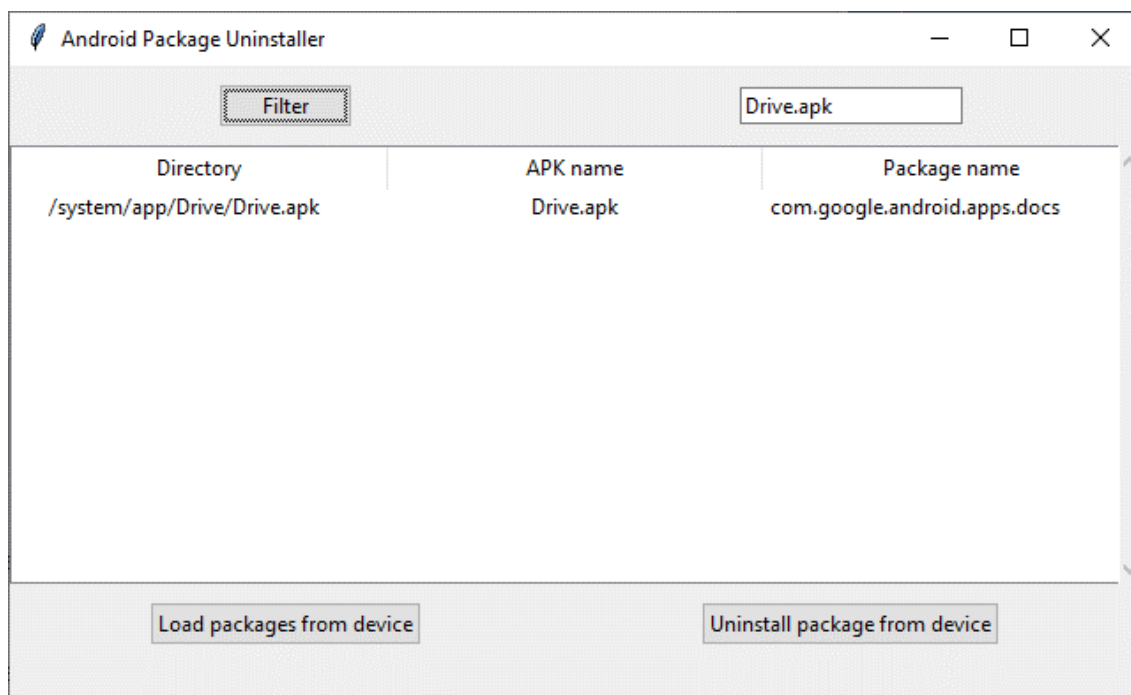


Ilustración 11.5 Ventana de la aplicación al filtrar los paquetes por un texto

IV. Análisis y conclusiones del trabajo

12. Conclusiones

Como se había mencionado al principio de este documento, este trabajo tiene como objetivo dar visibilidad sobre los problemas de la privacidad y más en concreto en los dispositivos móviles inteligentes, que cada vez tienen más presencia en el día a día. Además, mediante la aplicación desarrollada, se ha intentado dar solución de cierta manera para aquellos usuarios que utilicen dispositivos que tengan como sistema operativo Android. Es por ello que se ha hecho un plan de desarrollo y se han identificado ciertos riesgos a la hora de trabajar y desarrollar la aplicación, para los que se ha intentado buscar soluciones para que en caso de que algo saliese mal, se pudiese retomar el desarrollo lo antes posible.

En la Fase I se ha recogido la información actual sobre la recolección de los datos de los usuarios en los diferentes dispositivos de uso general a través de los sistemas que estos usan. Se ha visto como el simple hecho de tener aplicaciones en el sistema puede ser suficiente para la recolección de cierta información, además de que no siempre se es consciente de la información que se recopila desde cada compañía.

Además, se ha visto como no solo las compañías que hacen los sistemas operativos o aquellas que desarrollan servicios digitales recogen información, sino que las mismas compañías que ofrecen servicios físicos o que venden los propios dispositivos móviles recogen toda la información del usuario que consideran valiosa o que les pueden aportar grandes beneficios a la larga.

En la Fase II se ha contado cómo ha sido el desarrollo de la aplicación, explicando los conceptos necesarios para entender cómo se ha buscado una solución posible al problema de la recolección masiva de información en los dispositivos Android. Además, se ha buscado desarrollar la aplicación de la manera más modular y flexible posible ya que, al hacer la aplicación de código abierto, se busca que cualquiera pueda modificarla o mejorarla sin que sea necesario llegar a tocar la lógica de la aplicación. Para ello se ha hecho accesible en el siguiente repositorio de GitHub, donde cualquiera puede descargar el código y hacer peticiones de cambio o reportar incidencias:

- <https://github.com/Queky/androidpackageuninstaller>

Cabe destacar que la aplicación desarrollada cumple con el propósito inicial del trabajo, pero también es importante mencionar que se debe tener cierto conocimiento básico sobre qué es lo que está el usuario desinstalando, ya que, aunque el dispositivo se puede restaurar y no perder aplicaciones fundamentales para el funcionamiento del dispositivo, esto puede acabar causando un proceso bastante engorroso y la posible pérdida de información en el proceso de restauración.

12.1. Problemas durante el desarrollo

Durante el desarrollo de la aplicación se han ido encontrando ciertos factores que han influido en la estimación, y por lo tanto, en la dedicación necesaria para completar tareas. En ciertas ocasiones simplemente han sido decisiones a tomar antes de continuar el desarrollo, mientras que en otras ocasiones por culpa de estos factores se ha tenido que cambiar el enfoque y, por ende, cómo se atacaba el problema a solucionar.

Uno de los primeros problemas encontrados fue para qué plataforma desarrollar la aplicación. Esto no supuso un problema en sí, pero sí que hubo que garantizar que la base de código sobre la que se iba a desarrollar la aplicación fuese compatible con todos los entornos posibles (o, al menos, los entornos para los que está disponible ADB). Esto era importante por varias razones:

- Se quería desarrollar una aplicación de código abierto que pudiese ser ejecutada directamente desde su código fuente, sin necesidad de ejecutar un archivo generado con la aplicación
- El código fuente tenía que poder ser transformado en un ejecutable para los entornos más usados por los usuarios:
 - Windows (formato `.exe`)
 - macOS (aplicación Python)
 - Sistemas basados en Linux (aplicación Python)

Ya que Python resolvía todos estos problemas, se eligió como el lenguaje de programación sobre el que se iba a desarrollar la aplicación.

Esto trajo un problema indirecto entre el lenguaje de programación y la arquitectura elegida para el desarrollo de la aplicación: el uso de interfaces en Python resultó ser algo más complejo al principio de lo que se creía. Finalmente, esto se resolvió sin mayores problemas, ya que gracias a la documentación de Python se ha conseguido hacer uso de interfaces en Python.

Por último, otro de los problemas encontrados fue a la hora de desarrollar la parte gráfica de la aplicación. Ya que la idea era desarrollar una interfaz lo más ligera posible, y que fuese lo más flexible posible, se eligió Tkinter.

El principal motivo de elegir esta librería para la interfaz gráfica fue que se podía desarrollar programáticamente, por lo que no haría falta programas añadidos para el diseño de la misma. Al principio y para cosas sencillas sí que resultó útil, pero cuando se intentó desarrollar el listado de aplicaciones en el centro de la aplicación, se antojó algo difícil encontrar en la documentación la información necesaria para diseñar tal y como se quería ese componente.

Finalmente, y después de investigar en diferentes foros, se consiguieron resolver las dudas y por lo tanto desarrollar correctamente el componente.

12.2. Trabajo futuro

En cuanto al trabajo futuro, la aplicación todavía tiene mucho desarrollo posible ya que no se han explotado todas las funcionalidades que los paquetes de desarrollo de Android (no solo ADB, sino alguno más que viene incluido y que no se ha mencionado en las explicaciones de este trabajo) ofrecen.

Por ejemplo, sobre otro de los paquetes disponibles dentro de estas utilidades que proporciona Google, podemos encontrar *fastboot*²⁹. Esta es una librería que permite modificar las particiones del dispositivo, por lo que podríamos mediante esta librería modificar la partición *recovery* de un dispositivo, o flashear un nuevo sistema en el dispositivo. Este tipo de tareas son mucho más complejas ya que dependen mucho del dispositivo del usuario y del desarrollo logrado por la comunidad de Android, por lo que no sería una acción disponible de manera general, como sí lo es la aplicación desarrollada en este trabajo.

También se ha echado en falta una base de conocimiento sobre las aplicaciones, tanto sus nombres de paquetes como información de para qué sirven. Esto hace que la aplicación sea útil para la desinstalación de paquetes que el usuario ya conoce, pero no para el resto de paquetes que muchas veces, además, tienen un nombre poco descriptivo. Esta funcionalidad podría añadirse directamente a la desarrollada, dando toda esta información una vez el usuario selecciona uno de los paquetes del dispositivo.

Además, como se ha mencionado en alguna sección de este trabajo, esta misma aplicación podría enfocarse (en una nueva ventana, por ejemplo) a conseguir acceso root en el dispositivo que se conecte, facilitando todos los pasos al usuario y haciendo que fuese una simple acción desde un ordenador.

También podría enfocarse en la instalación de un *custom recovery*, que como ya se ha hablado en este documento, es un *recovery* que proporciona más funcionalidades al usuario.

Fuera ya de lo que corresponde a modificaciones en el dispositivo, también podría añadirse una funcionalidad para que hiciese una auditoria del dispositivo. Esta podría hacerse en base a los paquetes que tiene el dispositivo instalados, y para cada uno de ellos, los permisos y los accesos al diferente hardware que tienen.

Esta funcionalidad sería interesante porque, además, si se hiciese junto a la base de conocimiento anteriormente nombrada, podría decir a un usuario por ejemplo si una aplicación instalada en su dispositivo es legítima o no.

Por último, otro posible desarrollo para esta aplicación sería comprobar las aplicaciones que se están ejecutando en el dispositivo en segundo plano. Sería útil para identificar, por ejemplo, qué paquetes estarían quizá recolectando información cuando el usuario no se da cuenta.

²⁹ <https://www.androidcentral.com/android-z-what-fastboot>

V. Apéndices

Código fuente y documentación del proyecto

Este proyecto se ha desarrollado con la idea de que sea abierto al público, es decir, que sea un proyecto Open Source. Aunque su objetivo o enfoque esté claro y esté bastante acotado, el trabajo futuro a desarrollar ya tiene como base las herramientas con las que se puede continuar el desarrollo. Además, gracias a que se ha intentado desacoplar el código entre sí lo máximo posible, esta aplicación puede usarse tanto completa, como por separado, usando solo la parte funcional o la visual en una nueva aplicación.

Una de las razones por las que se ha optado por hacer público el código fuente de la aplicación es que el propio código fuente de Android es libre. Las herramientas usadas son las librerías de desarrollo que proporciona Google, y dentro de esta, más específicamente, *Android Debug Bridge*, que, aunque no son parte del sistema operativo Android, ni tampoco son Open Source, están disponibles para todos los usuarios que quieran usarlas.

El código fuente se encuentra en la siguiente URL, donde se encuentran ciertas aclaraciones de uso y se publicarán las versiones específicas para ejecutarse en cada uno de los entornos compatibles:

- <https://github.com/Queky/androidpackageuninstaller>

Bibliografía

- [1] Política de privacidad de Microsoft. 2020. URL: <https://privacy.microsoft.com/es-es/privacystatement>
- [2] The Guardian. *Windows 10 sends identifiable data to Microsoft despite privacy settings*. Ago. de 2015. URL: <https://www.theguardian.com/technology/2015/aug/13/windows-10-sends-identifiable-data-microsoft-despite-privacy-settings>
- [3] Política de privacidad de Apple. 2020. URL: <https://www.apple.com/legal/privacy/es/>
- [4] Política de privacidad de Google. 2020. URL: <https://policies.google.com/privacy?hl=es-ES#infocollect>
- [5] Declaración de privacidad de Ubuntu. 2020. URL: <https://ubuntu.com/legal/data-privacy>
- [6] Listado de compañías adquiridas por Facebook. 2020. URL: https://en.wikipedia.org/wiki/List_of_mergers_and_acquisitions_by_Facebook
- [7] Política de privacidad de Facebook. 2020. URL: <https://es-es.facebook.com/privacy/explanation>
- [8] Bernardmarr. *Amazon: Using Big Data to understand customers*. URL: <https://www.bernardmarr.com/default.asp?contentID=712>
- [9] Declaración de privacidad de Amazon. 2020. URL: <https://www.amazon.com/gp/help/customer/display.html?nodeId=GX7NJQ4ZB8MHFRNJ>
- [10] Xataka Android. *Comparamos a fondo las capas de personalización de Android: así es el software de Samsung, Huawei, LG, Xiaomi, Google y más*. Feb. de 2019. URL: <https://www.xatakandroid.com/sistema-operativo/comparamos-a-fondo-capas-personalizacion-android-asi-software-samsung-huawei-lg-xiaomi-google>
- [11] Política de privacidad de Samsung. 2020. URL: https://www.samsung.com/es/info/privacy_legal/
- [12] Xataka Android. *El veto de Estados Unidos a Huawei ya ha entrado en vigor: no habrá más prórrogas*. Ago. de 2020. URL: <https://www.xatakandroid.com/mercado/veto-estados-unidos-a-huawei-ha-entrado-vigor-no-habra-prorrogas>
- [13] Política de privacidad de Huawei. 2020. URL: <https://www.huawei.com/es/privacy-policy>
- [14] Edu Salguero. *Arquitectura Hexagonal*. Jun. de 2018. URL: <https://medium.com/@edusalguero/arquitectura-hexagonal-59834bb44b7f>
- [15] Wired. *The WIRED Guide to Your Personal Data (and Who Is Using It)*. Feb. de 2019. URL: <https://www.wired.com/story/wired-guide-personal-data-collection/>
- [16] Mashable. *Amazon Knows What You Want Before You Buy It*. Ene. de 2014. URL: <https://mashable.com/2014/01/21/amazon-anticipatory-shipping-patent/?europa=true>
- [17] Xataka. *Windows 10 y las mejoras iterativas: cuatro años después es mejor Windows que nunca, pero no es perfecto*. Mayo de 2019. URL: <https://www.xataka.com/aplicaciones/windows-10-mejoras-iterativas-cuatro-anos-despues-mejor-windows-que-nunca-no-perfecto>

- [18] The Verge. *Microsoft finally reveals what data Windows 10 really collects*. Abr. de 2017. URL: <https://www.theverge.com/2017/4/5/15188636/microsoft-windows-10-data-collection-documents-privacy-concerns>
- [19] Android Central. *Amazon Alexa: What kind of data does Amazon get from me?* Mar. de 2018. URL: <https://www.androidcentral.com/amazon-alexa-what-kind-data-does-amazon-get-me>
- [20] Android Debug Bridge. 2020. URL: <https://developer.android.com/studio/command-line/adb>
- [21] Cyber Security Hub. *Incident Of The Week: Garmin Pays \$10 Million To Ransomware Hackers Who Rendered Systems Useless*. Ago. de 2020. URL: <https://www.cshub.com/attacks/articles/incident-of-the-week-garmin-pays-10-million-to-ransomware-hackers-who-rendered-systems-useless>

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<https://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be

designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other

conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only

one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.