

DESARROLLO DE UNA HERRAMIENTA PARA EL CONTROL DE LA PRIVACIDAD EN ANDROID

IÑAKI SÁNCHEZ JIMÉNEZ

PABLO GONZÁLEZ NALDA

TRABAJO DE FIN DE GRADO

ÍNDICE

1. Requisitos funcionales
2. Planificación del tiempo
3. Conceptos generales
4. Recolección de datos
5. Tecnologías y herramientas
6. Desarrollo de la aplicación
7. Desarrollo de la aplicación
8. Implementación
9. Demo
10. Conclusiones

1. REQUISITOS FUNCIONALES

- ¿Qué esperamos de este trabajo y de la aplicación desarrollada?
 - Análisis de la privacidad de los usuarios
 - Una aplicación que nos muestre qué paquetes hay en el dispositivo
 - Una aplicación que nos permita eliminar cualquier paquete del dispositivo



2. PLANIFICACIÓN DEL TIEMPO

- Metodología ágil scrum

- Sprints de 2 semanas
- Tareas con un peso determinado en función de la dificultad
 - S (1)
 - M (3)
 - L (5)
- Si una tarea tiene un peso mayor, se divide

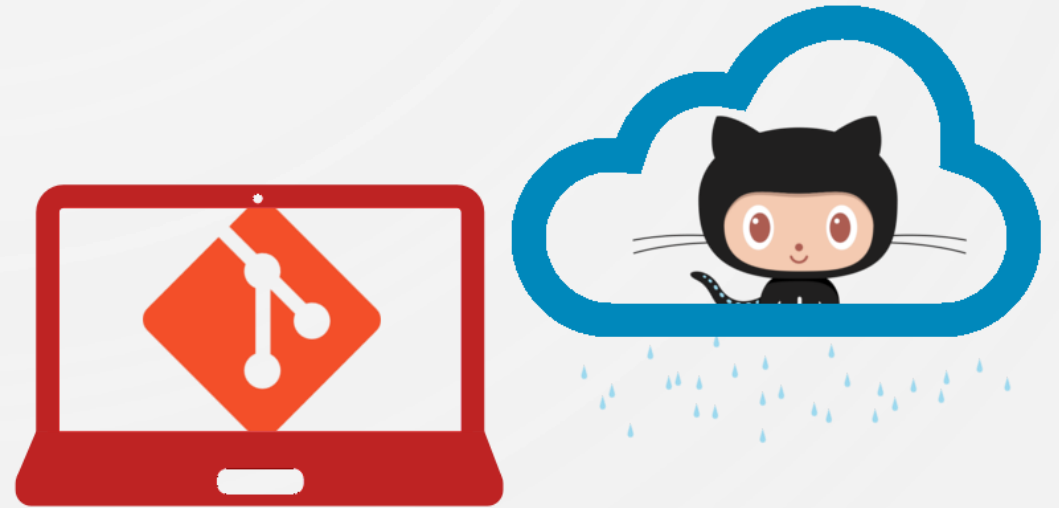
- Diferentes tipos de tareas

- Task
- Story
- Bug

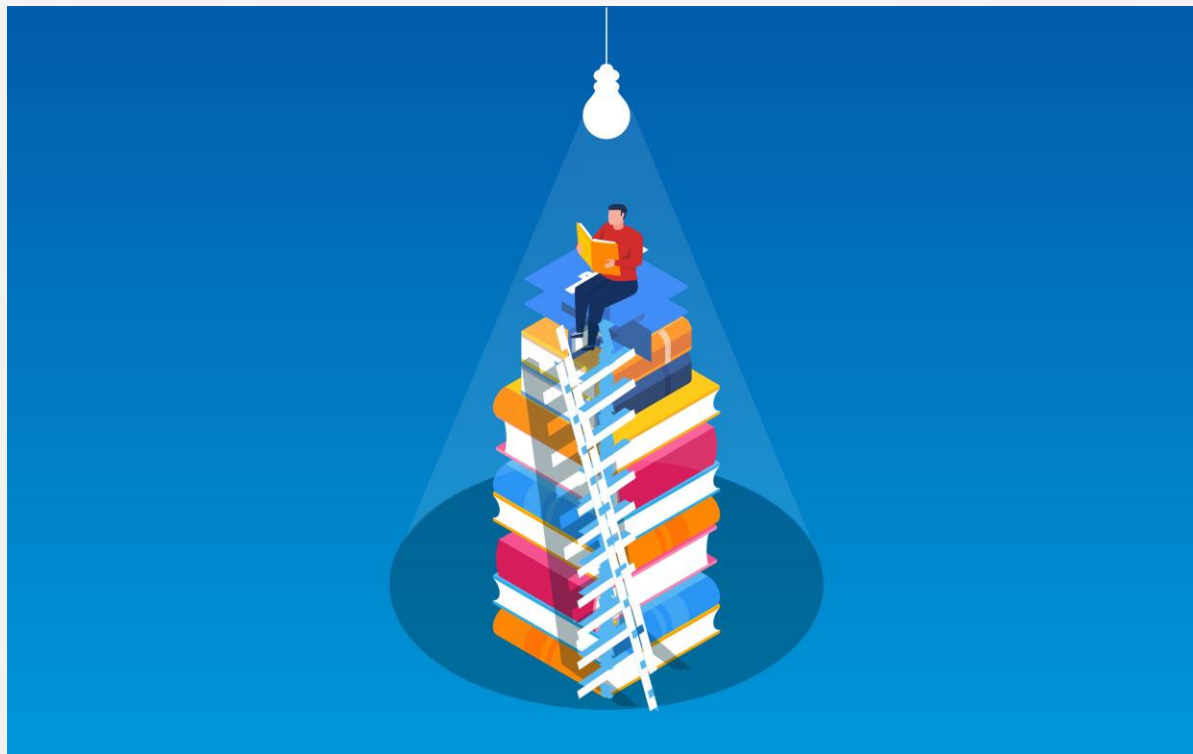


3. GESTIÓN DE RIESGOS

- Entorno de desarrollo fácilmente reproducible mediante “pip”
- Código guardado en la nube usando GitHub
- Uso de “git” para control de versiones



4. CONCEPTOS GENERALES



4.1 INFORMACIÓN RECOLECTABLE

- Medios por los que se puede recolectar información
- Tipo de información que se recolecta
- Cuando se recolecta esta información



4.2 USO DE LA INFORMACIÓN

- Mejoras en las aplicaciones
 - Detectar puntos débiles
 - Detectar puntos fuertes
 - Prevenir el fraude
- Catalogar usuarios
 - ¿Qué significa que me cataloguen?



4.3 SEGURIDAD DE LA INFORMACIÓN

- Privacidad vs Seguridad
- Información comprometida
 - ¿A quién afecta?



5. RECOLECCIÓN DE DATOS



5.1 ¿QUÉ DATOS SE RECOLECTAN?

- Datos personales
- Credenciales
- Datos demográficos
- Datos de pago
- Datos de consumo de contenido
- Búsquedas y comandos
- Comunicaciones
- Contenido multimedia



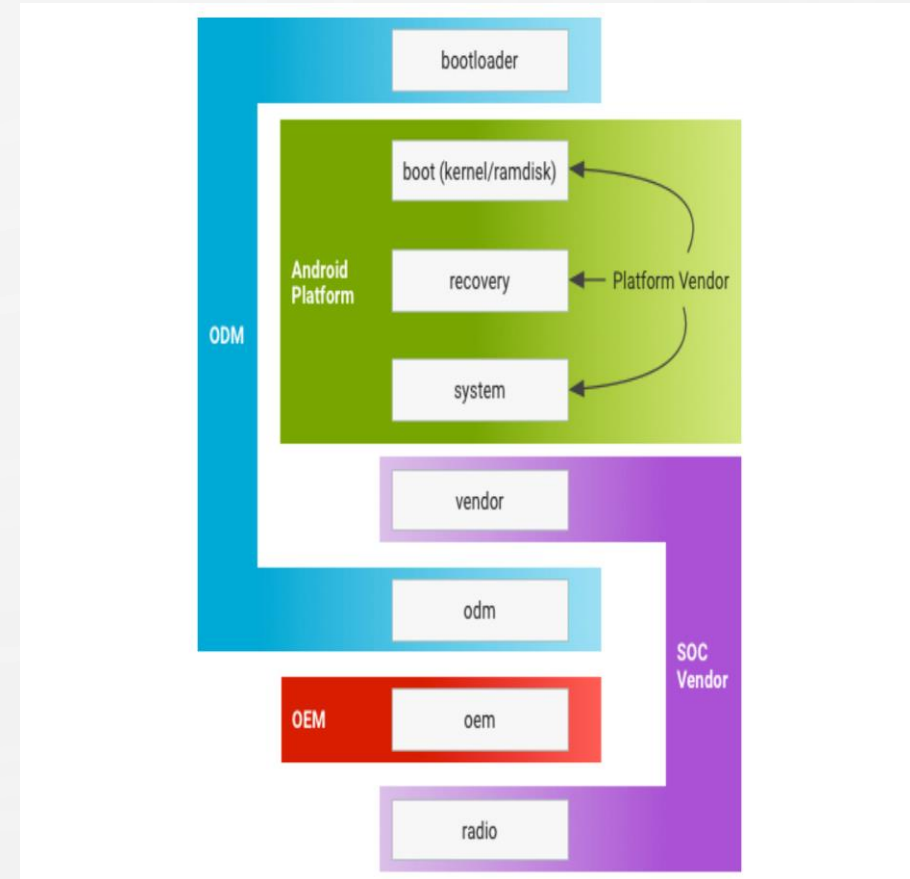
5.2 ¿PARA QUÉ SE RECOLECTAN?

- Para mejorar los productos existentes
- Desarrollar nuevos productos
- Personalizar los productos
- Hacer recomendaciones
- Para proporcionar publicidad personalizada
- Investigación

[illegible]

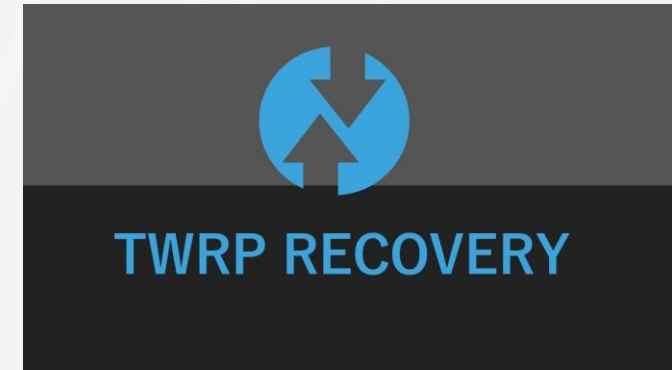
6.1 PARTICIONES DE ANDROID

- Particiones
 - Boot
 - Recovery
 - System



6.2 PARTICIÓN RECOVERY

- Custom recovery - ¿Qué nos permite hacer?
 - Backups
 - Instalación de aplicaciones con privilegios *root*
 - Acciones con privilegios sobre cualquier partición del dispositivo
 - Sobrescribir */system* para poder cambiar la distribución del dispositivo
 - Modificar la partición */boot* para cambiar el kernel del dispositivo
 - Para instalarlo necesitamos desbloquear el *bootloader*



6.3 MAGISK

- Magisk - ¿Qué nos permite hacer?
 - Obtiene acceso *root* en el dispositivo
 - Otorga acceso *root* a las aplicaciones que lo soliciten
 - Se tiene que instalar desde un *custom recovery*
 - Instalarlo supone la perdida de la garantía en el dispositivo



6.4 ANDROID DEBUG BRIDGE

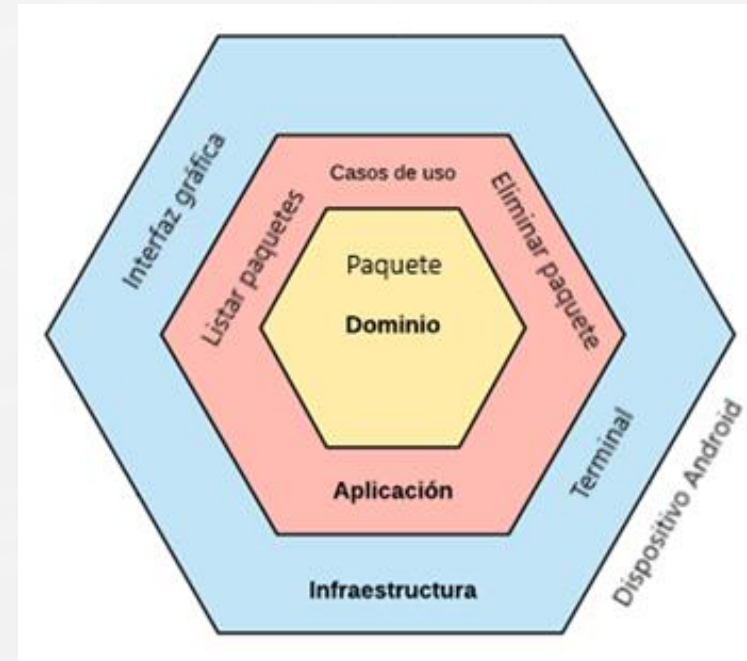
- Android Debug Bridge - ¿Qué nos permite hacer?
 - Distribuida con las *platform-tools* de Google
 - Herramienta de desarrollador
 - Se usa a través de la línea de comandos
 - Contiene comandos que se pueden enviar al dispositivo
 - Comando “*shell*”
 - Comando “*pm*” dentro del dispositivo

7. DESARROLLO DE LA APLICACIÓN



ARQUITECTURA

- Arquitectura hexagonal - ¿Qué nos ofrece?
 - Gran abstracción
 - Independencia entre módulos
 - Gran flexibilidad



CONEXIÓN CON EL DISPOSITIVO

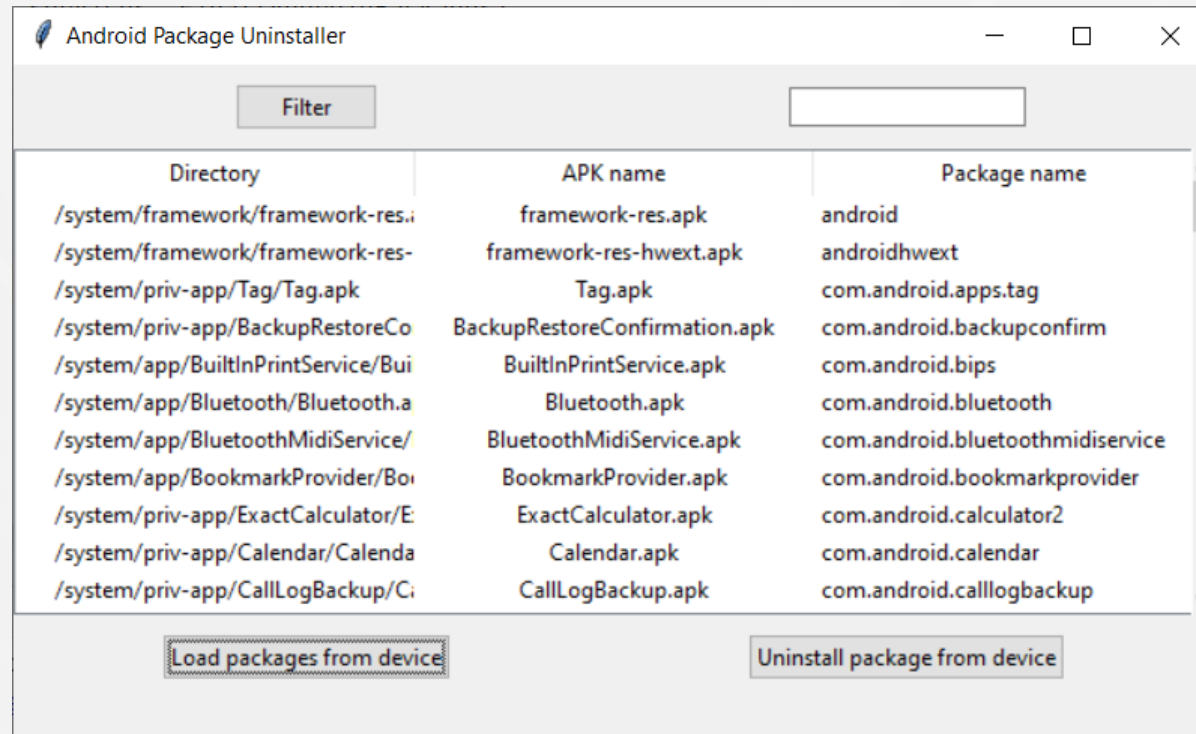
- Activando el modo desarrollo en el dispositivo
- A través de Python usando una librería propia
- Estos abstraen la terminal sin importar el entorno sobre el que se ejecuta la aplicación
- Cada entorno tiene comandos propios



FUNCIONALIDAD

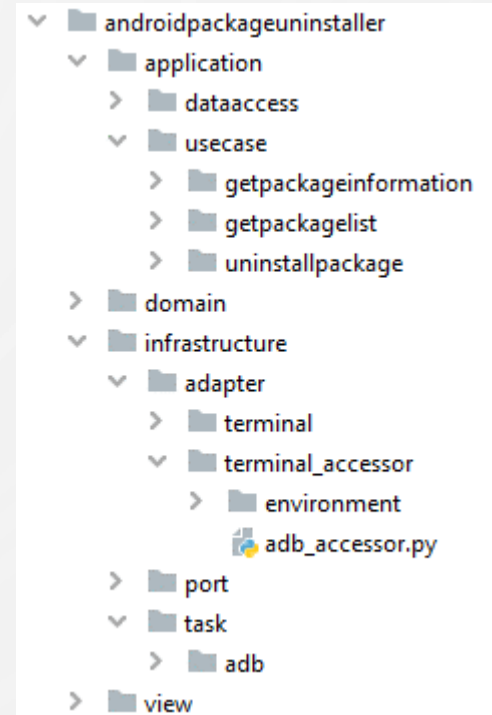
- Filtrar paquetes
- Listar todos los paquetes del dispositivo
- Desinstalar un paquete seleccionado

8. IMPLEMENTACIÓN



8.1 ESTRUCTURA DEL PROYECTO

- Cada capa de la aplicación tiene su modulo
- La conexión entre capas es mediante interfaces
- Solo las capas exteriores tienen conocimiento de las capas internas



8.2 INTEGRACIÓN Y USO DE ADB

- Es necesario descargar ADB
- Es un recurso externo para la aplicación
- Gracias a Python podemos abstraer el uso de la terminal en cualquier entorno
- Aunque la llamada para cada entorno sea diferente, la respuesta es la misma en todos

9. DEMO



10. CONCLUSIONES

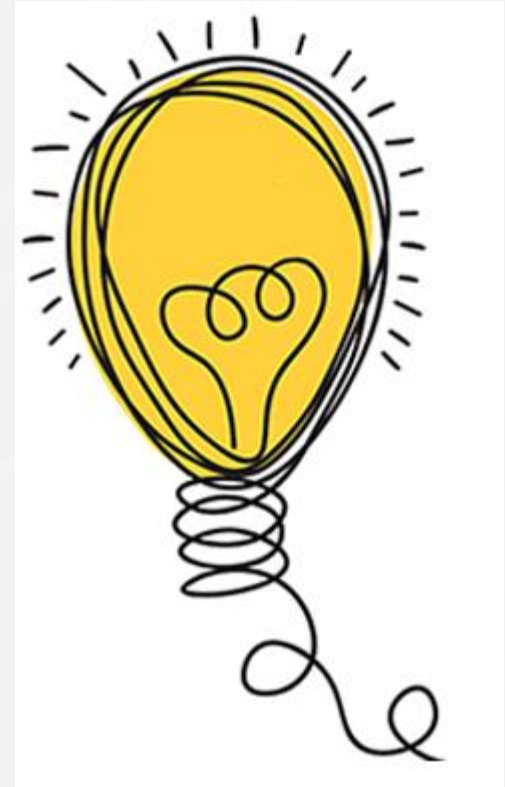


10.1 PROBLEMAS DURANTE EL DESARROLLO

- Hacer la aplicación multiplataforma
- Hacer una aplicación escalable
- Crear una aplicación propia para cada entorno
- Arquitectura con una fuerte dependencia en abstracciones
- Documentación de la librería de la interfaz grafica un poco pobre

10.2 TRABAJO FUTURO

- Información sobre los paquetes instalados
- Realizar una auditoria del dispositivo
- Usar la librería *fastboot* para otras funcionalidades
 - Instalar un *custom recovery*
 - Instalar una aplicación para ganar acceso *root*





¿PREGUNTAS?

¡Gracias!