DroidScrape

Suite de herramientas de inteligencia para Android

Alumno:
Mario de Benito Aspas

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Consultor:
Francesc D´Assís Giralt Queralt

Profesor Responsable de la Asignatura:
Carles Garrigues Olivella

7 de junio de 2017

FICHA DEL TRABAJO FINAL

Título del trabajo:	DroidScrape: Suite de herramientas de inteligencia para Android
Nombre del autor:	Mario de Benito Aspas
Nombre del consultor:	Francesc D´Assís Giralt Queralt
Nombre del profesor responsable:	Carles Garrigues Olivella
Fecha de entrega (mm/aaaa):	06/2017
Titulación:	*Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles*

Resumen del Trabajo (máximo 250 palabras):

El proyecto tiene como objetivo cubrir algunas de las necesidades del día a día de un Analista de Inteligencia sin necesidad de tener acceso a un ordenador.

Consta del desarrollo de un framework móvil escalable que permita la integración de diversos plugins que ofrezcan funcionalidades variadas relacionadas con las labores típicas de este servicio.

Además, se incluirán algunos plugins por defecto con herramientas como un analizador de cabeceras de correo electrónico y un *web scraper*.

Abstract (in English, 250 words or less):

This project aims to cover some of the needs presented in an Intelligence Analyst's daily life without the restrains caused by being forced to have access to a computer.

It features the development of a scalable mobile framework which allows the integration of different plugins that offer functionalities related to thes kind of service's common tasks.

Furthermore, some default plugins will be included. These plugins will offer some common tools such as an email header analyzer and a web scraper.

Palabras clave (entre 4 y 8):

Android, OSINT, framework, analysis, intelligence, tools

Índice

Lista de figuras

Lista de tablas

1 Introducción

En este apartado se explican las razones por las que se ha decidido llevar a cabo este proyecto, así como las principales decisiones tomadas para su ejecución.

1.1 Contexto y justificación del Trabajo

En la industria de la seguridad informática y la inteligencia en fuentes abiertas existe una necesidad continua de herramientas que faciliten el trabajo de los analistas a la hora de afrontar las amenazas que día a día aparecen en la red.

Por otra parte, es básico que este tipo de servicios se mantengan alerta en un horario 24x7, los 365 días del año. Esto significa que los analistas y especialistas que deberán responder ante una amenaza requieren una disponibilidad total de dichas herramientas, ya sea en su puesto de trabajo o fuera de él.

Si se centra el foco en los servicios de Ciberinteligencia y Vigilancia Digital, se puede afirmar que estamos hablando de una industria joven y poco conocida. Este punto se hace más evidente si observamos el panorama español del sector, donde existen pocas compañías que ofrezcan este tipo de servicios.

Quizá debido a estos dos puntos, la juventud del sector y el desconocimiento generalizado de la existencia de estos servicios, no existe un gran número de herramientas especializadas que cubran las necesidades de los servicios corporativos. Por ello, la mayoría de estos servicios se ven obligados a desarrollar herramientas ad-hoc para muchas de las tareas que no pueden cubrir con las herramientas generalistas existentes.

Si se extrapola esta falta de herramientas especializadas al mundo móvil, encontramos que son prácticamente inexistentes.

Las guardias y la disponibilidad constante

Si se tiene en cuenta todo lo explicado anteriormente, se puede entender que hay dos formas de ofrecer un servicio de calidad:

- Rotación de turnos de especialistas
- Guardias y periodos de disponibilidad

En el primer caso, es necesario que siempre haya un número suficiente de especialistas disponibles en la sede del servicio. Esto implica turnos nocturnos y otro tipo de horarios poco atractivos para los trabajadores que, además, en muchos casos, debido a su estatus de especialistas, no están dispuestos a cubrir.

En el segundo caso, si bien debe haber personal trabajando en la sede del servicio durante las 24 horas del día, los especialistas pueden tener un horario más acorde a su categoría profesional y cubrir el resto de horarios mediante guardias y disponibilidades.

Sin embargo, en este supuesto, los trabajadores que deben estar disponibles fuera de su horario normal, necesitan acceso a herramientas que puedan cubrir sus necesidades. Esto, debido a la falta de herramientas móviles orientadas a este sector, les obliga a llevar consigo un ordenador portátil o a mantenerse cerca de su lugar de trabajo.

Contexto tecnológico

En los últimos años, las capacidades técnicas de dispositivos móviles como *smartphones* y *tablets* han aumentado drásticamente. A pesar de que multitud de expertos consideran la Ley de Moore obsoleta, aparentemente esto no se puede aplicar aún a este tipo de dispositivos[1,2].

La mayoría de herramientas utilizadas por un servicio de inteligencia OSINT requieren de una gran capacidad de procesamiento y almacenamiento de datos. Estas capacidades aún están muy lejos del alcance de los dispositivos móviles, sin embargo, si los resultados de esas herramientas son correctamente filtrados, análisis más concretos y precisos pueden ser realizados en cualquier dispositivo.

Parece una propuesta interesante la creación de herramientas de utilidad para los especialistas de este sector que puedan ser ejecutadas desde sus dispositivos de bolsillo y les permitan por tanto un mayor grado de libertad cuando deben trabajar desde fuera de su puesto habitual.

1.2 Objetivos del Trabajo

El objetivo del proyecto es crear una *suite* de herramientas de inteligencia para dispositivos móviles, cubriendo así la necesidad de los trabajadores del sector del análisis de inteligencia de acceder a este tipo de recursos desde fuera de su lugar habitual de trabajo.

Para definir un alcance asequible y realista, se han definido los siguientes requisitos de alto nivel:

a) Framework que permita la escalabilidad: Deberá ser posible incluir nuevas herramientas en la aplicación sin un gran esfuerzo. La aplicación deberá por tanto tener una base modular que permita añadir herramientas en futuras actualizaciones sin afectar a las anteriormente desarrolladas.

b) Módulo de análisis de cabeceras de correo: Deberá incluir un módulo que permita al usuario analizar cabeceras de correo de la forma más exhaustiva posible.

c) Módulo de *web scraping*: Deberá incluir un módulo que permita al usuario realizar un scraping de un sitio web en busca de unos términos concretos.

d) Publicación en market oficial: La aplicación deberá quedar publicada en el market oficial de Google.

1.3 Enfoque y método seguido

En este apartado se explica el proceso que se pretende llevar a cabo para alcanzar los hitos del proyecto y cumplir en la mayor medida posible los objetivos establecidos en el punto anterior.

Debido a la falta de herramientas móviles que cubran los objetivos establecidos y tomando como partida la experiencia personal del desarrollador en el sector, se realizará una herramienta ad-hoc con las funciones necesarias tomando como base algunas existentes para otras plataformas.

Para poder cumplir el objetivo **a**, será necesario empezar por desarrollar un *framework* que permita ir añadiendo módulos o *plugins* que cubran el resto de objetivos funcionales o nuevos requisitos que puedan surgir una vez terminado el desarrollo inicial.

Una vez el *framework* esté preparado, se procederá a desarrollar los distintos *plugins* que dotarán a la herramienta de funcionalidad.

Análisis de herramientas existentes

Se tomarán como ejemplo para los distintos módulos herramientas existentes para otras plataformas:

- Módulo de análisis de cabeceras de correo: MxToolbox, IP2Location
- Módulo de web scraping: Scrapy

Un análisis de estas y otras herramientas similares se puede encontrar en el Anexo I: Análisis de herramientas similares.

Licencia

Con el objetivo de que la aplicación sea lo más accesible posible, se realizará con una licencia que permita su reutilización y modificación: **GNU GPLv3**[3].

Gracias a este tipo de licencia, individuos ajenos al proyecto tendrán la capacidad de desarrollar nuevos plugins de utilidad creando una aplicación mantenida por la comunidad.

Plataforma y tecnología

Esta herramienta, previsiblemente, ejecutará tareas costosas con relativa frecuencia. Es por esta razón que se ha descartado el desarrollo de una aplicación híbrida o una aplicación web, cuyo rendimiento es mucho menor que el de una aplicación nativa.

La plataforma elegida para el proyecto será Android. Esta decisión se basa en dos puntos:

- Android es un sistema de Código Abierto, algo que encaja con la filosofía del proyecto.
- Android domina el mercado en cuanto a implantación (Ilustración 1).

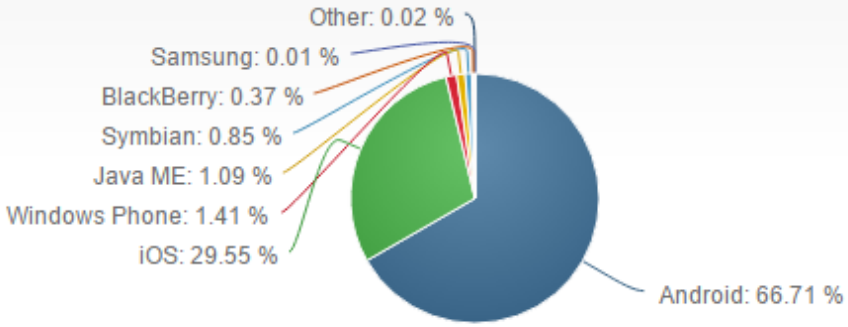


Ilustración 1: Distribución del mercado en móviles y tabletas entre distintos sistemas operativos en febrero del 2017[4]

Tras analizar la cuota de mercado de cada una de las versiones de este Sistema Operativo, será un requisito que la aplicación pueda funcionar en una versión 4.4 (KitKat) o superior para garantizar su compatibilidad en el mayor número posible de dispositivos (Ilustración 2).

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	3.7%
4.2.x		17	5.4%
4.3		18	1.5%
4.4	KitKat	19	20.8%
5.0	Lollipop	21	9.4%
5.1		22	23.1%
6.0	Marshmallow	23	31.3%
7.0	Nougat	24	2.4%
7.1		25	0.4%

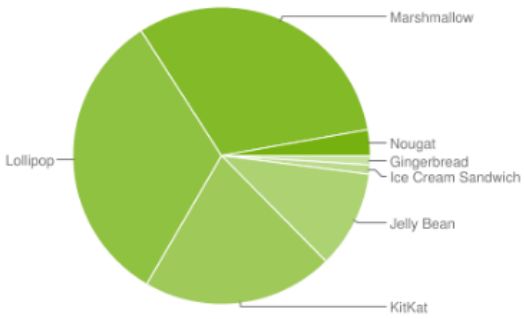


Ilustración 2: Distribución de versiones de Android en el mercado a 6 de marzo de 2017[5]

1.4 Planificación del Trabajo

En este apartado se analiza la planificación del tiempo y los recursos necesarios para llevar a cabo el desarrollo.

Recursos técnicos necesarios para el desarrollo

- IDE: Android Studio
- Emulador para Android: Genymotion
- Repositorio Git privado
- Licencia de desarrollador de Google para la publicación de la aplicación

Planificación temporal

Para acotar el proyecto en el tiempo, se definen diversos hitos y fechas límite (Tabla 1).

Código	Hito	Fecha límite
H1	Plan de trabajo del proyecto	15/03/2017
H2	Diseño del proyecto	05/04/2017
H3	Implementación	17/05/2017
H4	Documentación y publicación en beta	07/06/2017
H5	Presentación de la herramienta	23/06/2017
H6	Mejoras derivadas de la fase beta	31/07/2017
H7	Publicación de la release	31/08/2017

Tabla 1: Hitos del proyecto

Para conseguir alcanzar cada uno de estos hitos, se definen una serie de tareas (Tabla 2).

Código de Hito	Código de Tarea	Tarea	Duración (horas)	Dedicación (horas/día)
H1	H1T1	Elección de proyecto	2	2
H1	H1T2	Análisis de requisitos	6	2
H1	H1T3	Análisis de contexto y herramientas existentes	4	2
H1	H1T4	Planificación	8	2
H1	H1T5	Documentación	4	2
H2	H2T1	Diseño de pantallas	12	3
H2	H2T2	Diseño estructural del framework	6	3
H2	H2T3	Diseño del plugin de análisis de cabeceras	6	3
H2	H2T4	Diseño del plugin de web scraping	5	2,5
H3	H3T1	Desarrollo del framework	15	3
H3	H3T2	Desarrollo del plugin de análisis de cabeceras	28	3
H3	H3T3	Desarrollo del plugin de web scraping	12	3
H3	H3T4	Integración de resultados de los plugins con el framework	8	2
H3	H3T5	Testing y resolución de bugs: módulo de análisis de cabeceras	8	2
H3	H3T6	Testing y resolución de bugs: módulo de web scraping	8	2
H4	H4T1	Documentación del proyecto: memoria	14	2
H4	H4T2	Documentación del proyecto: manual de usuario	14	2
H4	H4T3	Generación de contenido gráfico (iconos, screenshots...)	4	2
H4	H4T4	Publicación en Google Play en beta	3	3
H5	H5T1	Preparación de la presentación de la herramienta	20	2
H6	H6T1	Análisis de resultados y feedback de la beta	12	1
H6	H6T2	Mejoras y bugfixing	-	2
H7	H7T1	Publicación en Google Play (release)	3	3

Tabla 2: Tareas del proyecto

Para una mayor claridad, se incluye un diagrama de Gantt que describe la planificación del desarrollo del proyecto (Ilustración 3).

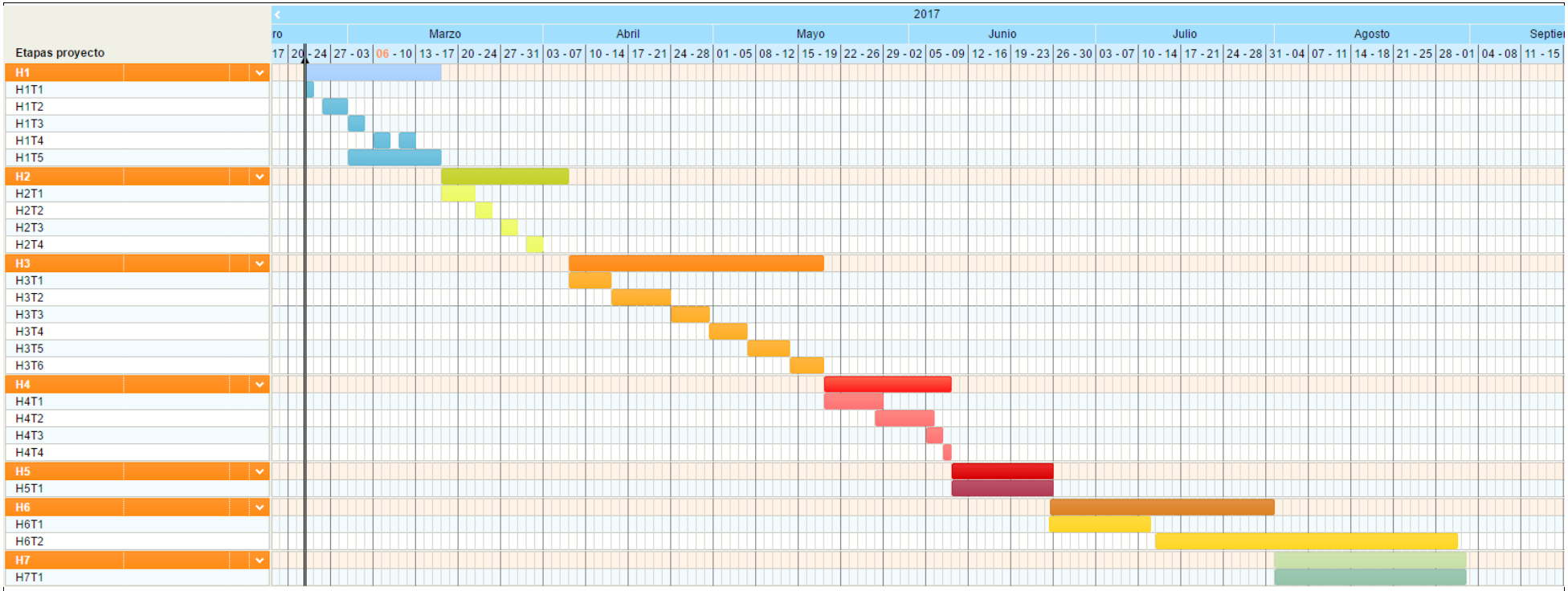


Ilustración 3: Diagrama de Gantt del proyecto

Criterios de aceptación del proyecto

A continuación se definen los criterios mínimos para que se considere que el proyecto ha alcanzado un nivel de desarrollo aceptable.

1. Cumplimiento del hito H3 al completo: El objetivo del proyecto es crear una herramienta funcional que cubra la necesidad de un usuario objetivo. Por ello, para que el proyecto alcance un nivel aceptable, deberá haberse producido una aplicación móvil que cumpla los requisitos de la herramienta.
2. Cumplimiento del hito H5: El proyecto se desarrolla en el marco de un Trabajo Fin de Máster, por lo que para que se pueda considerar que su desarrollo se ha llevado a cabo con éxito, deberá ser presentado satisfactoriamente en el ámbito y condiciones establecidas para tal trabajo.

Priorización de tareas

Para facilitar la actuación ante posibles riesgos, se han asignado prioridades a las tareas siguiendo el siguiente baremo:

- Tareas críticas (C): El no completar esta tarea en tiempo y forma puede suponer el fracaso del proyecto
- Tareas importantes (I): Estas tareas son básicas para que el nivel de calidad del proyecto sea óptimo
- Tareas secundarias (S): Estas tareas aportan calidad al proyecto pero este podría ser aceptable sin ellas

En la Tabla 3 se pueden consultar las prioridades asignadas a las tareas.

Código de Hito	Código de Tarea	Tarea	Prioridad
H1	H1T1	Elección de proyecto	C
H1	H1T2	Análisis de requisitos	C
H1	H1T3	Análisis de contexto y herramientas existentes	I
H1	H1T4	Planificación	C
H1	H1T5	Documentación	I
H2	H2T1	Diseño de pantallas	C
H2	H2T2	Diseño estructural del framework	C
H2	H2T3	Diseño del plugin de análisis de cabeceras	C
H2	H2T4	Diseño del plugin de web scraping	C
H3	H3T1	Desarrollo del framework	C
H3	H3T2	Desarrollo del plugin de análisis de cabeceras	C
H3	H3T3	Desarrollo del plugin de web scraping	C
H3	H3T4	Integración de resultados de los plugins con el framework	C
H3	H3T5	Testing y resolución de bugs: módulo de análisis de cabeceras	I
H3	H3T6	Testing y resolución de bugs: módulo de web scraping	I
H4	H4T1	Documentación del proyecto: memoria	C
H4	H4T2	Documentación del proyecto: manual de usuario	I
H4	H4T3	Generación de contenido gráfico (iconos, screenshots...)	I
H4	H4T4	Publicación en Google Play en beta	I
H5	H5T1	Preparación de la presentación de la herramienta	C
H6	H6T1	Análisis de resultados y feedback de la beta	S
H6	H6T2	Mejoras y bugfixing	S
H7	H7T1	Publicación en Google Play (release)	S

Tabla 3: Priorización de las tareas

Análisis de riesgos para el cumplimiento de la planificación

En este apartado se analizarán algunos riesgos que se deberán tener en cuenta a la hora de mantener la planificación del proyecto.

Código de riesgo	Riesgo	Medidas preventivas	Medidas reactivas
R1	Imprevistos que impidan continuar el desarrollo de forma temporal	Priorización de tareas dentro de cada Hito	Replanificación de tareas atendiendo a su prioridad
R2	Imprevistos que impidan continuar con el desarrollo por un periodo largo de tiempo	Planificación acorde a las prioridades	Flexibilización de fechas límite Replanificación del proyecto en siguiente periodo disponible
R3	Problemas técnicos o de disponibilidad de recursos	Copias de seguridad remotas Disponibilidad de varios equipos de trabajo	Traspaso del proyecto a una infraestructura de respaldo

Tabla 4: Resumen de riesgos para el cumplimiento de la planificación

R1. Imprevistos que impidan continuar el desarrollo de forma temporal:

Pueden surgir situaciones imprevistas que provoquen el cese de la actividad del proyecto de forma temporal, por ejemplo, enfermedad, excesiva carga de trabajo ajeno al proyecto o movilidad geográfica por razones ajenas al proyecto. Debido a que el equipo de desarrollo se compone únicamente de una persona, este tipo de imprevistos retrasarían toda la planificación establecida.

a) Medidas preventivas: Se establecen una serie de tareas prioritarias.
b) Medidas reactivas: Se deberán replanificar las tareas pendientes para evitar que tareas críticas queden fuera de plazo o tengan asignado un periodo de tiempo insuficiente.
c) Relevancia:

Impacto / Prob.	Improbable	Poco probable	Bastante probable	Muy probable
Bajo				
Moderado				
Grave		**X**		
Catastrófico				

R2. Imprevistos que impidan continuar con el desarrollo por un periodo largo de tiempo

Situaciones en las que surgiera algún tipo de imprevisto grave que impidiese continuar con el desarrollo del proyecto por un periodo suficientemente largo como para imposibilitar el alcance de alguno de los hitos considerados necesarios para la aceptación del proyecto en su tiempo límite. Se trata de un riesgo similar al R1 pero que implique un parón en el desarrollo que no sea posible recuperar.

a) Medidas preventivas: Planificación acorde a las prioridades, intentando que las tareas más críticas sean, en la medida de lo posible, ejecutadas antes que las menos críticas.
b) Medidas reactivas: En caso de materializarse este riesgo, se deberían intentar renegociar las fechas límite y, en caso de no ser posible, se postpondrá la ejecución del proyecto al próximo periodo disponible.
c) Relevancia:

Impacto / Prob.	Improbable	Poco probable	Bastante probable	Muy probable
Bajo				
Moderado				
Grave				
Catastrófico		**X**		

R3. Problemas técnicos o de disponibilidad de recursos

Situaciones en las que alguno de los recursos necesarios para llevar a cabo el desarrollo no esté disponible por alguna razón, por ejemplo, por rotura de un equipo informático o caducidad de una licencia de software.

a) Medidas preventivas: Mantenimiento de copias de seguridad en la nube y repositorios remotos con control de cambios. Disponibilidad de varios equipos preparados para continuar con el desarrollo.
b) Medidas reactivas: Traspaso del desarrollo del proyecto a una infraestructura de respaldo.
c) Relevancia:

Impacto / Prob.	Improbable	Poco probable	Bastante probable	Muy probable
Bajo		**X**		
Moderado				
Grave				
Catastrófico				

Cálculo de valor del proyecto

A continuación se incluye un cálculo del valor del proyecto en horas de trabajo y recursos técnicos.

En el caso de los recursos técnicos, se analiza si ya se dispone de ellos y, en caso contrario, se especifica la inversión necesaria (Tabla 5).

Recurso	¿Disponible?	Valor	Inversión necesaria
Infraestructura informática	Sí	800€	0€
Infraestructura de respaldo	Sí	600€	0€
Licencia de desarrollador de Google	Sí	25€	0€
IDE	Sí	0€	0€
Emulador	Sí	0€	0€
Repositorio remoto	Sí	0€	0€

Tabla 5: Valoración de recursos técnicos

En cuanto al valor en horas de trabajo, se utilizará como referencia un salario de 30.000€ anuales (15,6€/hora).

En la Tabla 2 se estableció un total de 202 horas de trabajo para llevar el proyecto a término, con lo cual, la valoración del proyecto ascendería a 3.353,2€ en cuanto a horas de trabajo.

1.5 Breve sumario de productos obtenidos

En cada una de las fases del desarrollo, que terminan con la consecución de un hito, se generan una serie de entregables (Tabla 6).

Hito	Entregables
H1	Documentación: Plan de trabajo del proyecto
H2	Documentación: Diseño de pantallas Documentación: Diseño estructural y diagrama de clases
H3	Código: Apk funcional
H4	Documentación: Memoria del proyecto Documentación: Manual de usuario Documentación: Recursos gráficos Publicación: Fase beta
H5	Documentación: Feedback preliminar de fase beta Documentación: Presentación de la herramienta Documentación: Vídeo de presentación
H6	Código: Apk actualizado con mejoras derivadas de la beta Documentación: Changelog
H7	Publicación: Release

Tabla 6: Entregables del proyecto

1.6 Breve descripción de los otros capítulos de la memoria

2 Diseño de la aplicación

En este capítulo se detalla la información relevante que se ha obtenido al realizar el análisis y diseño de la aplicación.

3 Funcionamiento del Framework

En este apartado se detalla el funcionamiento del framework diseñado.

4 Pruebas unitarias

En este capítulo se explican las diversas pruebas unitarias incluidas con el proyecto.

5 Resultados de la evaluación

Una explicación de los resultados producidos por las distintas fases de evaluación del proyecto.

6 Cambios en el diseño

En este capítulo se describen los principales cambios que el proyecto ha sufrido desde su diseño inicial hasta su versión final.

7 Conclusiones y agradecimientos

Agradecimientos a personas que han colaborado en el proyecto y algunas conclusiones extraídas del trabajo en el mismo.

8 Próximos pasos

En este capítulo se proponen algunas posibles ampliaciones y mejoras del proyecto que se realizarán en un futuro.

9 Glosario

En este apartado se aportan definiciones y explicaciones de algunos términos específicos del sector de la inteligencia y la seguridad o de palabras poco comunes utilizadas a lo largo de la memoria.

10 Bibliografía

En esta sección se aporta la información bibliográfica utilizada a lo largo del documento.

11 Anexos

Los anexos incluyen distinta información relevante para el proyecto que, por su extensión o por su relación no directa con el mismo, se ha incluido de forma independiente al documento principal.

2 Diseño de la aplicación

En este capítulo se detalla la información relevante que se ha obtenido al realizar el análisis y diseño de la aplicación.

2.1 Usuarios y contexto de uso

Con el objetivo de definir las funcionalidades que deberá tener la aplicación final, en este apartado se analizan las características de los usuarios, sus necesidades y sus objetivos.

Descripción del usuario objetivo

El usuario objetivo de la aplicación es un profesional de la seguridad informática o del sector de la inteligencia que está acostumbrado a la utilización de software específico del sector.

Debido a la usual complicación de las interfaces de las herramientas típicas del sector, este usuario está acostumbrado a ver gran cantidad de datos en pantalla sin que el diseño visual sea una de sus prioridades.

Por otra parte, sus conocimientos técnicos son avanzados, por lo que la aplicación no necesitará aportar información de ayuda para todos aquellos términos específicos del sector o términos técnicos relacionados con las tecnologías de la información.

La razón por la que este usuario utiliza la aplicación es tener las herramientas que utiliza comúnmente en su trabajo fuera de la oficina integradas en un solo entorno. Para él es importante poder introducir información de forma sencilla, así como extraer los resultados de la aplicación para su envío por correo.

También es importante que la aplicación opere, en la medida de lo posible, mediante datos móviles en ausencia de otra conexión más estable. Por ello, se deberá tratar de optimizar cualquier intercambio de datos con servidores de Internet.

Debido a la delicadeza de los datos tratados por la aplicación, que pueden ser confidenciales, tras cada ejecución no deberá quedar rastro en el dispositivo de los mismos.

Perfil técnico	Avanzado
Manejo de dispositivos móviles	Avanzado
Percepción del diseño visual	Poco relevante
Movilidad	Alta
Confidencialidad de los datos	Alta

Tabla 7: Resumen de características del usuario objetivo

Personas y perfiles de ejemplo

Para ilustrar más claramente el perfil objetivo del proyecto, se han generado varias Personas que representan a potenciales usuarios de la aplicación detallando su contexto y de qué manera podría serles útil DroidScrape.

- Persona 1: Samuel
  - Cargo: Analista experto en seguridad
  - Nivel técnico: muy elevado
  - Contexto: Samuel tiene que estar siempre disponible para resolver dudas técnicas de sus compañeros y atender peticiones urgentes para análisis de amenazas. Una de las peticiones más frecuentes es el análisis de cabeceras de emails relacionados con campañas de phishing o spam. Para realizar dicho análisis se ve obligado a tener cerca su ordenador, lo que le impide alejarse de casa mientas está de guardia.
  - Necesidades cubiertas por la aplicación: La aplicación le permitirá realizar un análisis rápido de las cabeceras que le permita enviar información relevante a sus compañeros y decidir si merece la pena llevar a cabo una investigación más en profundidad.
- Persona 2: Iván
  - Cargo: Operador de vigilancia digital
  - Nivel técnico: medio
  - Contexto: Iván trabaja en turnos en un servicio de detección de amenazas. Como parte de sus funciones, debe atender peticiones de clientes interesados en recibir información ampliada sobre una amenaza que se ha detectado. Si la amenaza es de un carácter técnico elevado, contacta con sus compañeros especialistas de guardia. Sin embargo, Iván está interesado en la seguridad y en proceso de aprendizaje, por lo que le gustaría participar más activamente en la resolución de este tipo de casos. Su problema es que en su puesto no tiene acceso a herramientas apropiadas para realizar los análisis requeridos ya que su acceso a Internet está limitado.
  - Necesidades cubiertas por la aplicación: La aplicación le permitirá realizar un análisis previo que le permita seguir de cerca y entender las conclusiones que más adelante emitirá el analista experto. También le permitirá proporcionar más datos a este analista facilitándole su trabajo.

Funcionalidades contempladas en el alcance inicial

Con el objetivo de acotar el proyecto para su primera versión publicada, se comenzará ofreciendo únicamente dos funcionalidades como ya se explicó en el punto 1.2 Objetivos del Trabajo.

Dichas funcionalidades serán las siguientes:

- Analizador de cabeceras de correo
- Scraper web

Para cumplir con las necesidades explicadas en el apartado anterior, se especifican los siguientes requisitos:

- El usuario deberá poder introducir fácilmente una cabecera de correo en la aplicación, a ser posible, de forma directa desde alguno de los clientes más utilizados para Android. Para cubrir la posibilidad de que el usuario utilice algún cliente no compatible o no disponga de un cliente de correo conocido, será un requisito obligatorio que la cabecera de correo se pueda introducir desde el portapapeles.
- La aplicación deberá producir un informe sobre la cabecera web que deberá poder ser enviado por correo con el cliente elegido por el usuario.
- Los datos relativos a dicha cabecera de correo, tanto la entrada como los resultados, deberán ser eliminados al cerrar la aplicación.
- El usuario podrá introducir una url y una serie de términos para operar con el scraper web. Estos datos podrán ser tecleados o introducidos desde el portapapeles.
- El scraper web deberá producir un informe que contenga un listado de urls y los términos localizados en las mismas. Este informe deberá poder ser enviado directamente por correo con el cliente que seleccione el usuario. También deberá poder ser copiado en el portapapeles.
- Los datos relativos a la operación del scraper, tanto la entrada como los resultados, deberán ser eliminados al cerrar la aplicación.
- Los informes producidos por las herramientas deberán ser lo suficientemente claros como para ser de utilidad a personas con conocimientos técnicos y del sector pero que no tengan conocimiento de la aplicación.

2.2 Diseño conceptual

En este apartado, se presentan los casos de uso de la aplicación.

El flujo común de eventos que lleva a un usuario objetivo a utilizar la aplicación es el siguiente:

1. Un peticionario, normalmente un cliente o un analista de nivel inferior al usuario, remite a este una petición para realizar una investigación sobre un ítem (cabecera de correo o sitio web) para conseguir un objetivo concreto (localizar al remitente, determinar si hay contenido relevante en un sitio web).
2. El usuario utiliza las herramientas de la aplicación para obtener información y obtiene los resultados.
3. El usuario puede realizar un análisis de los resultados, generar un informe y mandar la información al peticionario o puede remitir los resultados al analista para que sea él quien los analice y elabore el informe.

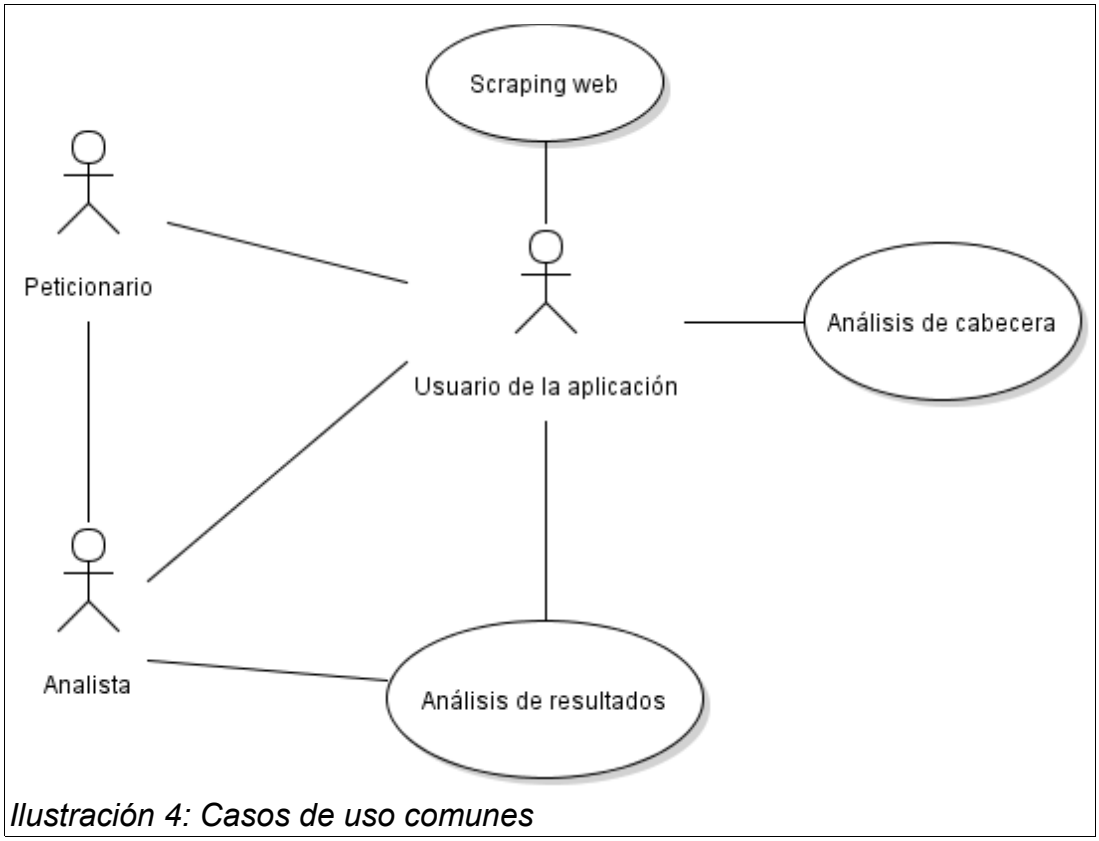


Ilustración 4: Casos de uso comunes

Actores	Precondiciones	Postcondiciones
Peticionario Analista Usuario de la aplicación	El usuario recibe una petición de análisis	El usuario dispone de un informe con los resultados del análisis

Sin embargo, y para tener en cuenta un caso más genérico, el usuario puede encontrarse realizando una investigación y necesitar de alguna de las herramientas contempladas en la aplicación. En este caso, el flujo sería el siguiente:

1. El usuario introduce la información relevante en la aplicación.
2. Obtiene los resultados de operar con la herramienta.
3. Analiza los resultados para incorporarlos a su investigación.

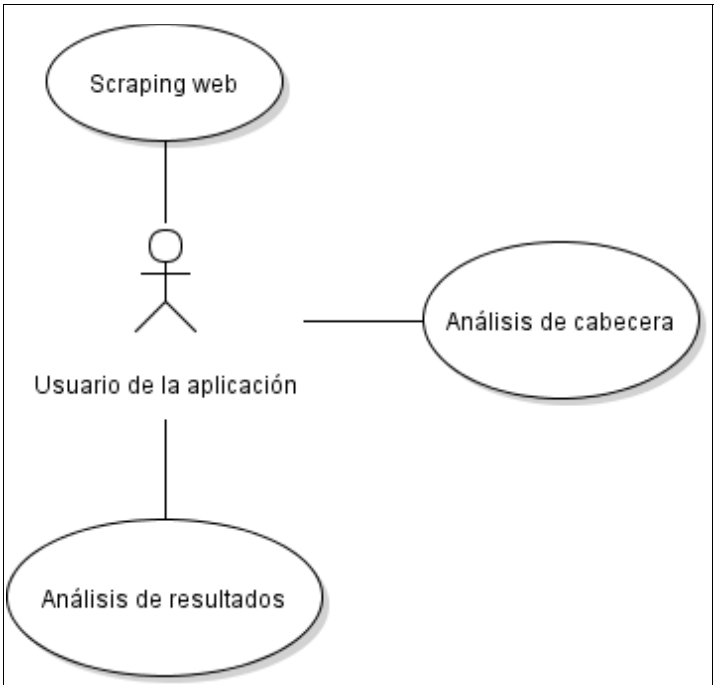


Ilustración 5: Casos de uso genéricos

Actores	Precondiciones	Postcondiciones
Usuario de la aplicación	El usuario tiene la necesidad de realizar un análisis para su investigación	El usuario dispone de un informe con los resultados del análisis

Si tenemos en cuenta el primer caso, en el que intervienen actores que no tienen por qué tener conocimiento de la aplicación, es necesario que la salida de la misma, es decir, el informe de resultados, sea legible y fácilmente entendible. De esta forma, los resultados obtenidos de operar con la herramienta serán útiles y no exigirán de un análisis extra por parte del usuario a la hora de ser compartidos con otros actores.

2.3 Prototipado

En este apartado se presentan los prototipos de las principales pantallas de la aplicación.

1. Menú principal (Ilustración 6): Esta pantalla se mostrará al iniciar la aplicación. En ella se mostrarán en una cuadrícula los distintos plugins existentes en la aplicación.
   https://wireframe.cc/zVuPaX

2. Analizador de cabeceras de email, introducción de datos (Ilustración 7): Será la pantalla inicial al acceder al plugin de análisis de cabeceras de email. En esta pantalla se mostrará un campo de introducción de texto para insertar los datos de la cabecera, un botón que permita rellenar dicho campo desde el portapapeles y un último botón que dará paso al análisis de los datos introducidos.
   https://wireframe.cc/Qunu5k

3. Analizador de cabeceras de email, resultados del análisis (Ilustración 8): Esta pantalla se mostrará tras completar el análisis de la cabecera de email introducida en la pantalla anterior. Incluirá un botón que permita copiar el informe en el portapapeles y otro que permita compartirlo mediante email o mensajería instantánea.
   https://wireframe.cc/nUrj7X

4. Web scraper, introducción de datos (Ilustración 9): Será la pantalla principal de este plugin y contendrá todos los campos necesarios para configurar el escaneo (URL, nivel de profundidad, agresividad del escaneo, términos de cerca), un botón para lanzarlo y uno para limpiar el formulario.
   https://wireframe.cc/kRfO2J

5. Web scraper, escaneo en progreso (Ilustración 10): Esta pantalla se mostrará mientras el escaneo esté ejecutándose. En ella se mostrará información sobre el progreso del escaneo y un botón que permitirá detenerlo en cualquier momento, dirigiendo al usuario a la pantalla de resultados del escaneo.
   https://wireframe.cc/OQ0P6x

6. Web scraper, resultados del escaneo (Ilustración 11): Esta pantalla mostrará un resumen de los resultados del escaneo ejecutado. Incluirá un botón que permita copiar el informe en el portapapeles y otro que permita compartirlo mediante email o mensajería instantánea.
   https://wireframe.cc/2m8qsj

En todas estas pantallas se presentará, en la parte superior, una barra de título que indique en qué plugin se está trabajando, un botón que permita volver a la pantalla anterior y un botón de menú que permitirá volver al menú principal.

Todos los campos y objetos presentados deberán ser adaptables de forma que el uso de la aplicación esté optimizado en dispositivos de distintas resoluciones y tamaños (móviles, tabletas...)

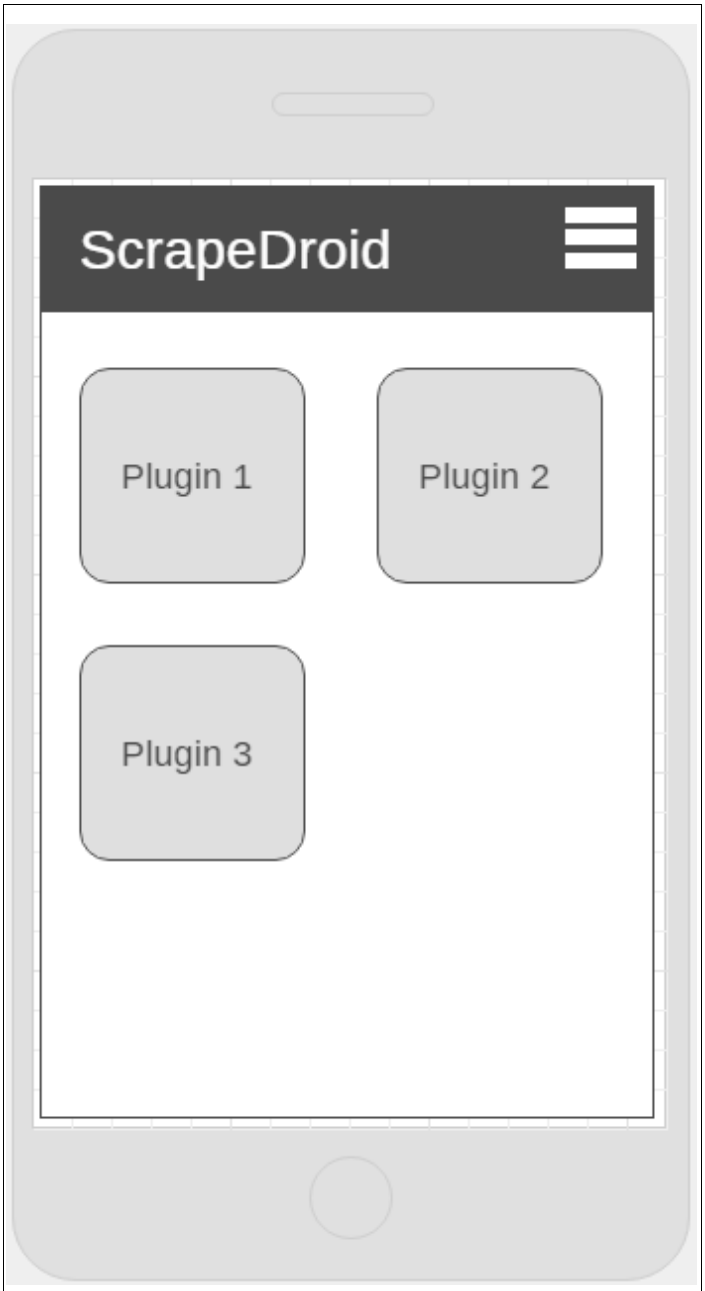


Ilustración 6: Pantalla 1 - Menú principal

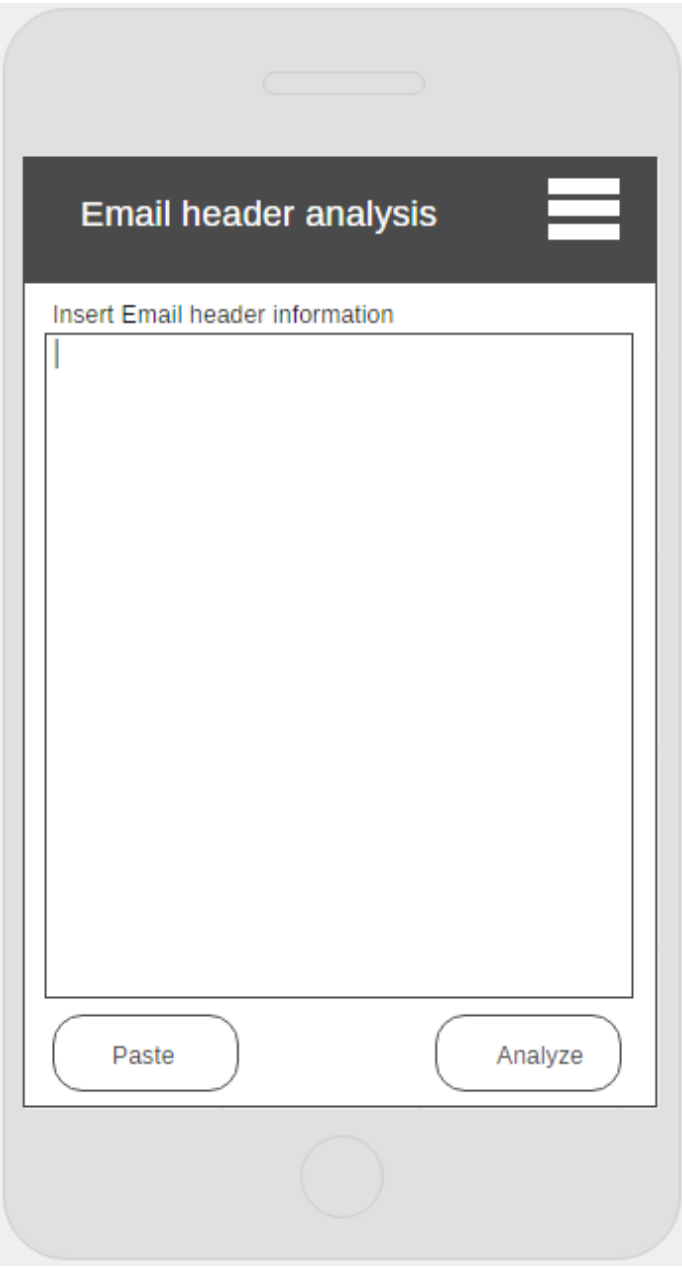


Ilustración 7: Pantalla 2 - Analizador de cabeceras de email: introducción de datos

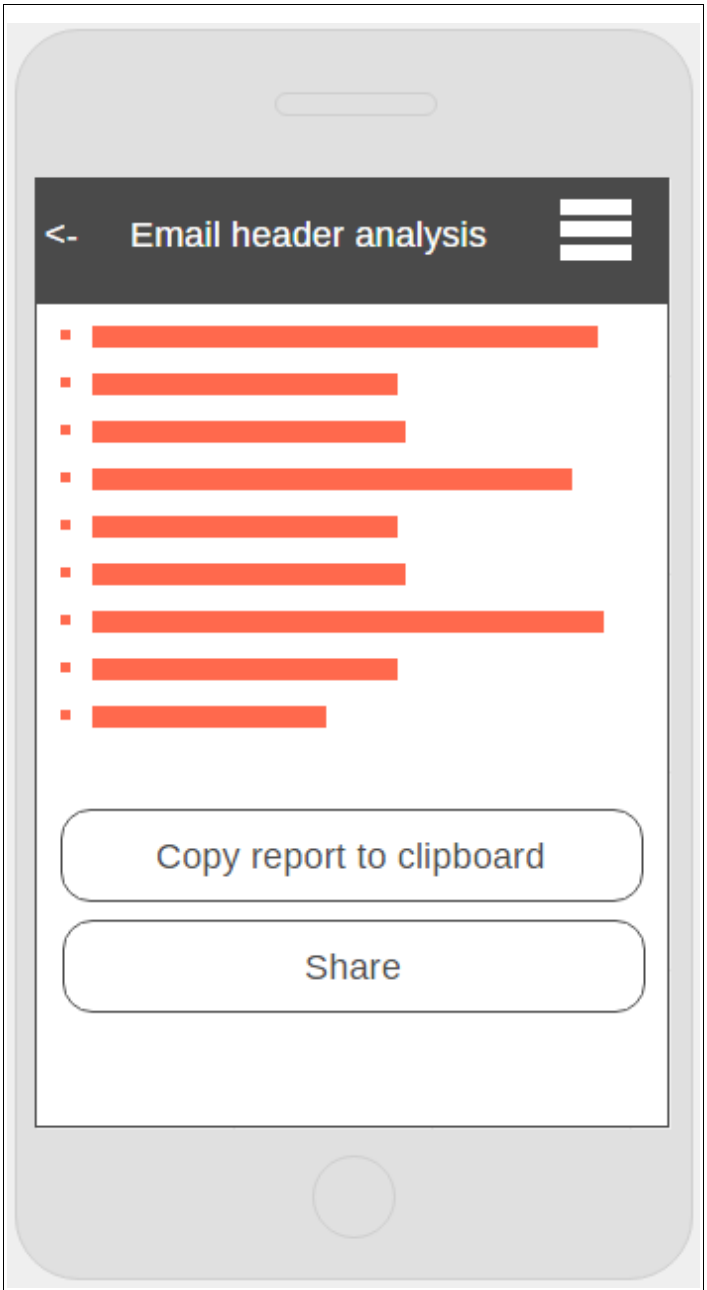


Ilustración 8: Pantalla 3 - Analizador de cabeceras de email: resultados del análisis

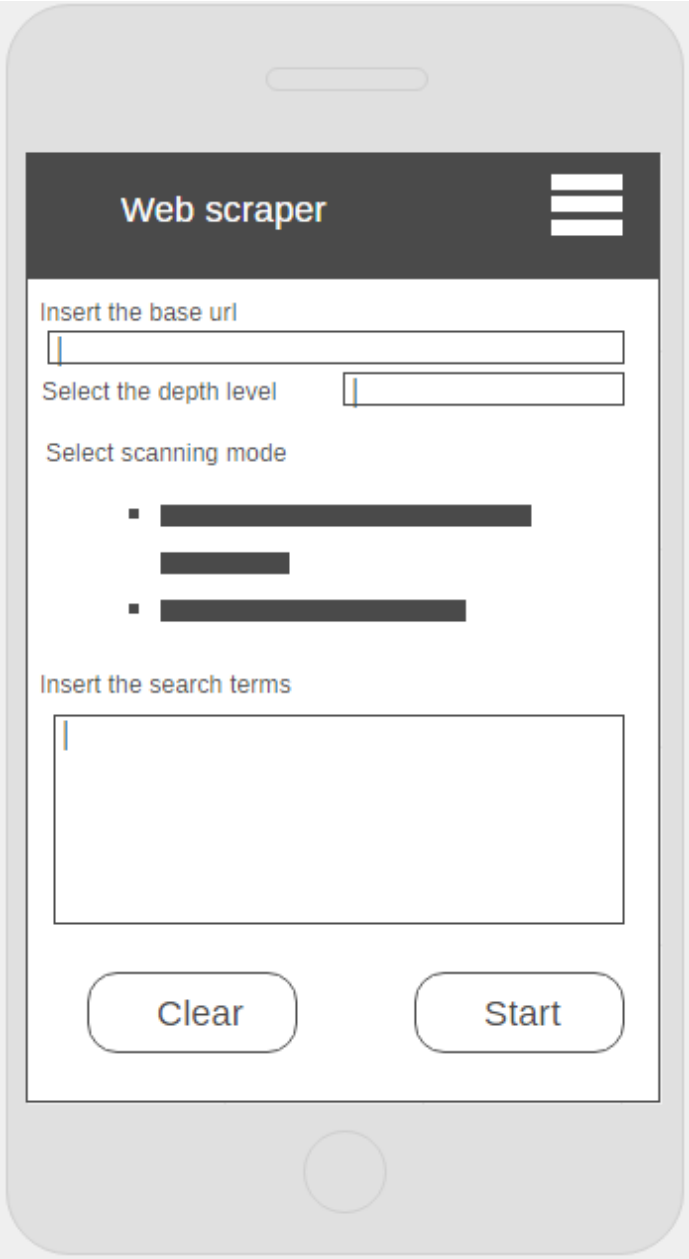


Ilustración 9: Pantalla 4 - Web scraper: introducción de datos

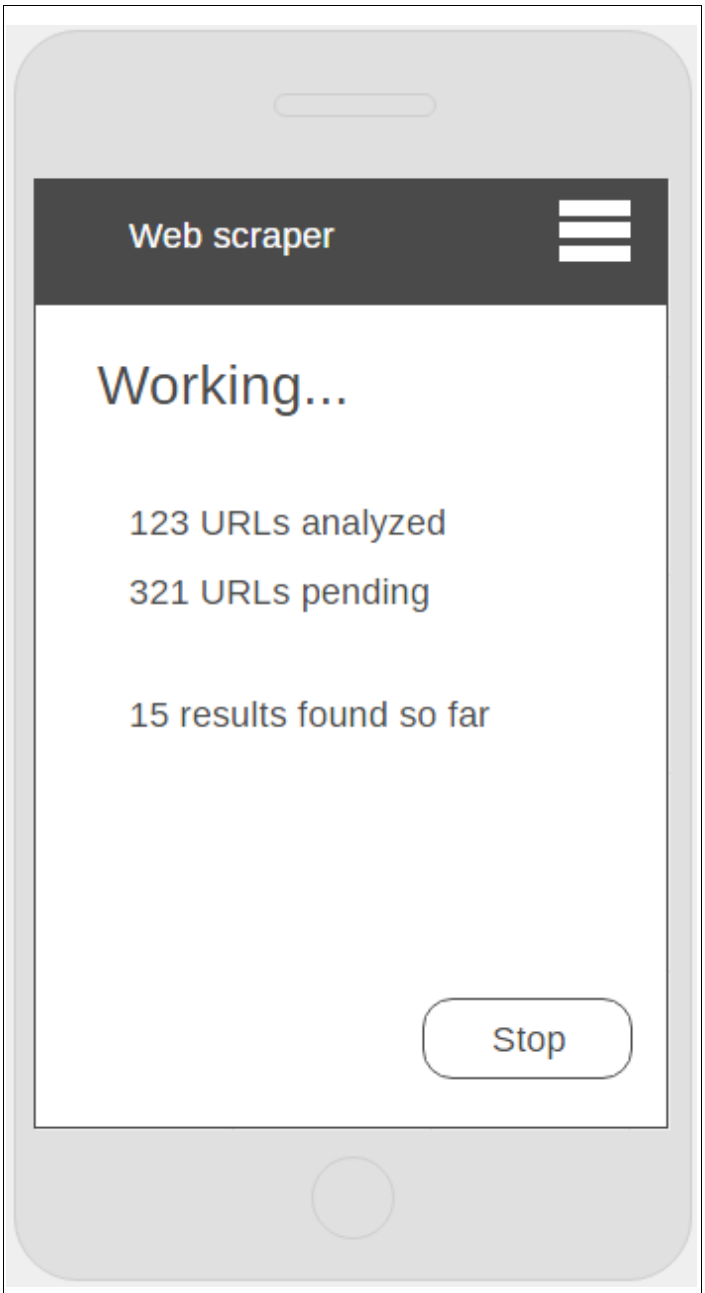


Ilustración 10: Pantalla 5 - Web scraper: escaneo en progreso

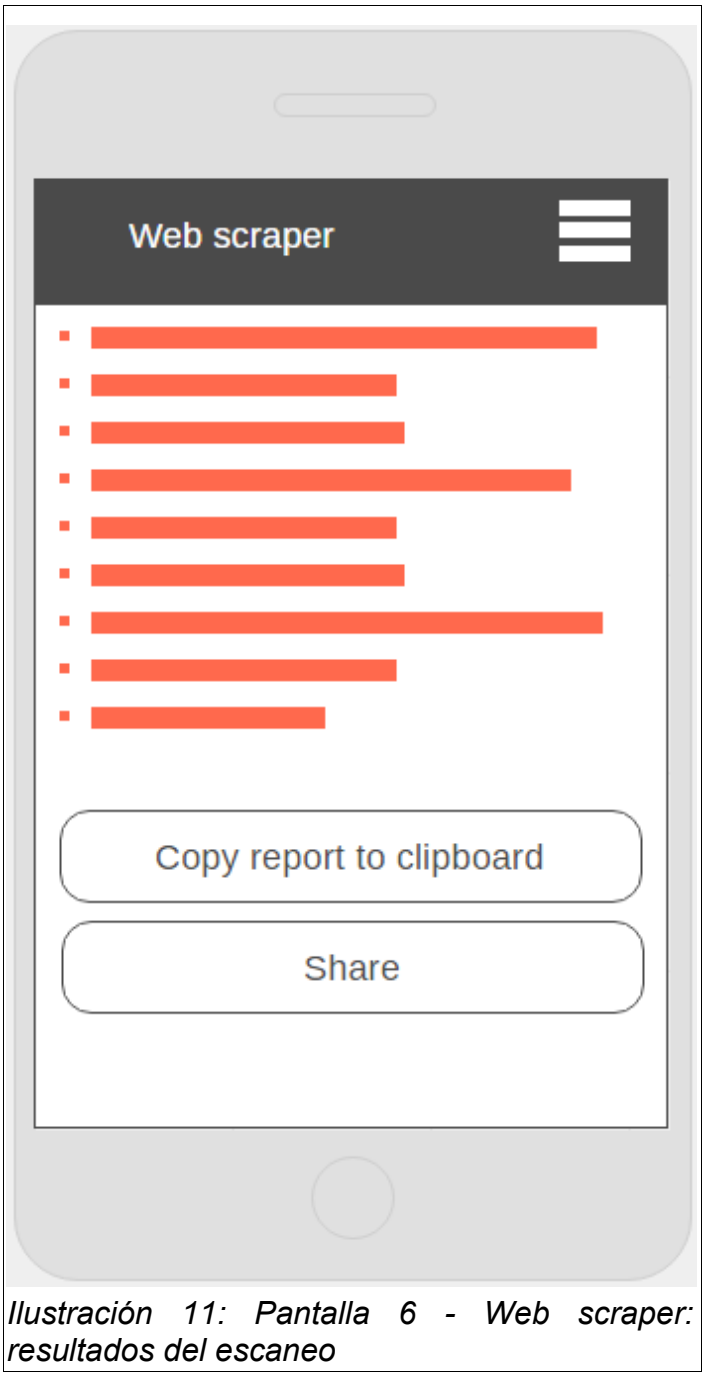


Ilustración 11: Pantalla 6 - Web scraper: resultados del escaneo

2.4 Evaluación

En este apartado se propone una estrategia de evaluación del diseño para la mejora del mismo de forma iterativa.

Debido a las especiales características de esta aplicación, donde el desarrollo del código que ejecutará los análisis cobra especial importancia, la evaluación del diseño de la interfaz se realizará en paralelo con el desarrollo de la aplicación.

Se divide dicha evaluación en tres fases:

1. Evaluación de alto nivel: se presentan los prototipos de alto nivel a posibles usuarios objetivo de la aplicación para que propongan cambios.
2. Evaluación de prototipo funcional: se presenta una aplicación en la que se ha desarrollado únicamente la interfaz para que posibles usuarios objetivo propongan cambios.
3. Evaluación de la aplicación en beta: se proporcionará acceso a usuarios objetivo a la descarga en fase beta de la aplicación en Google Play y se recogerán sus impresiones.

Debido a la arquitectura de la aplicación, diseñada utilizando el patrón MVC, y a la simplicidad del diseño de su interfaz -requiere de un pequeño número de pantallas con características muy definidas-, los cambios necesarios en la interfaz descubiertos durante el proceso de evaluación no afectarán al proyecto en su conjunto debido al encapsulamiento de la capa de Vista.

La evaluación de la aplicación completa se llevará a cabo durante la fase de beta, correspondiente al Hito H4 y planificada a partir del 7 de junio de 2017 (Ver Tabla 2).

2.5 Diseño de la arquitectura

En este apartado se muestra el diseño propuesto para la arquitectura de la aplicación.

Arquitectura del sistema

La aplicación se desarrollará siguiendo un patrón MVC (Modelo-Vista-Controlador) que permitirá aislar las diferentes capas y hacer la aplicación fácilmente extensible. Además, cada uno de los módulos o plugins incluidos, estará también encapsulado de forma que no influya en el funcionamiento de los demás o en el de la aplicación principal.

Se trata de una aplicación *standalone*, es decir, que no depende de otros servicios externos para su funcionamiento y todos los componentes requeridos por ella deben ir incluidos en su paquete. Sin embargo, se desarrollará de tal forma que resulte sencillo incluir nuevas funcionalidades o plugins con pocas líneas de código.

Se distribuirá el código de la aplicación en varias capas:

- Vista: Layouts
- Aplicación principal: Ejercerá de controlador para la carga inicial de la aplicación
- Modelo: Estructura de datos de la aplicación principal
- Módulo de plugins: Existirá un gestor de plugins que realizará la carga de los mismos para entregar a la Aplicación principal la información necesaria para ejecutarlos. Por otra parte, cada plugin tendrá sus propios vista y modelo.

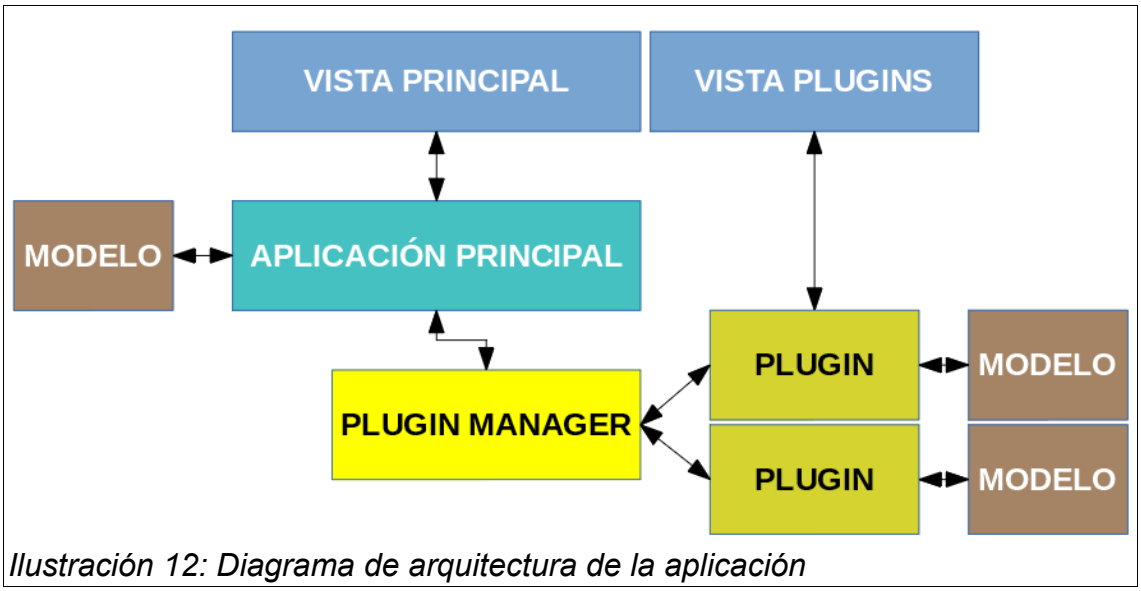


Ilustración 12: Diagrama de arquitectura de la aplicación

Los plugins irán en su propio paquete, dentro del paquete del framework y serán cargados al inicio de la aplicación. Para su carga, se utilizarán ficheros de configuración en formato JSON que se incluirán en los recursos de la aplicación.

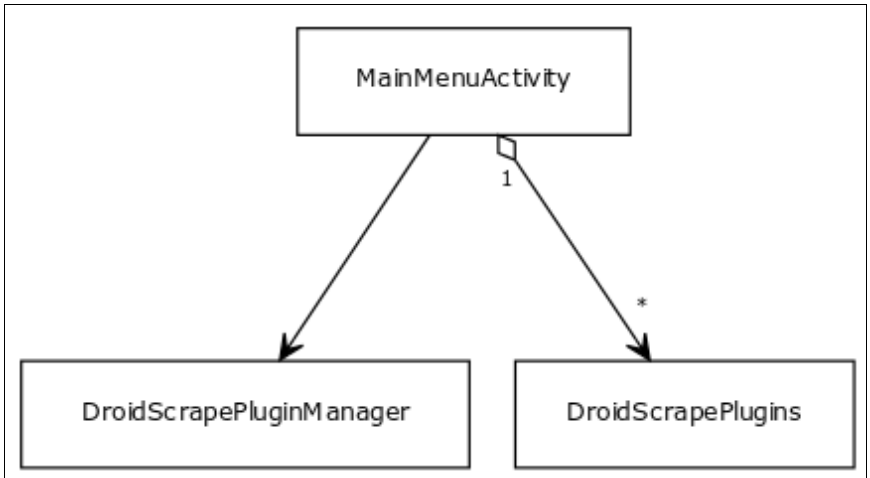


Ilustración 13: Diagrama de clases - Actividad principal

Existirá una clase específica llamada *DroidScrapePluginManager* que será la encargada de cargar los plugins instalados para que la aplicación principal pueda ejecutarlos.

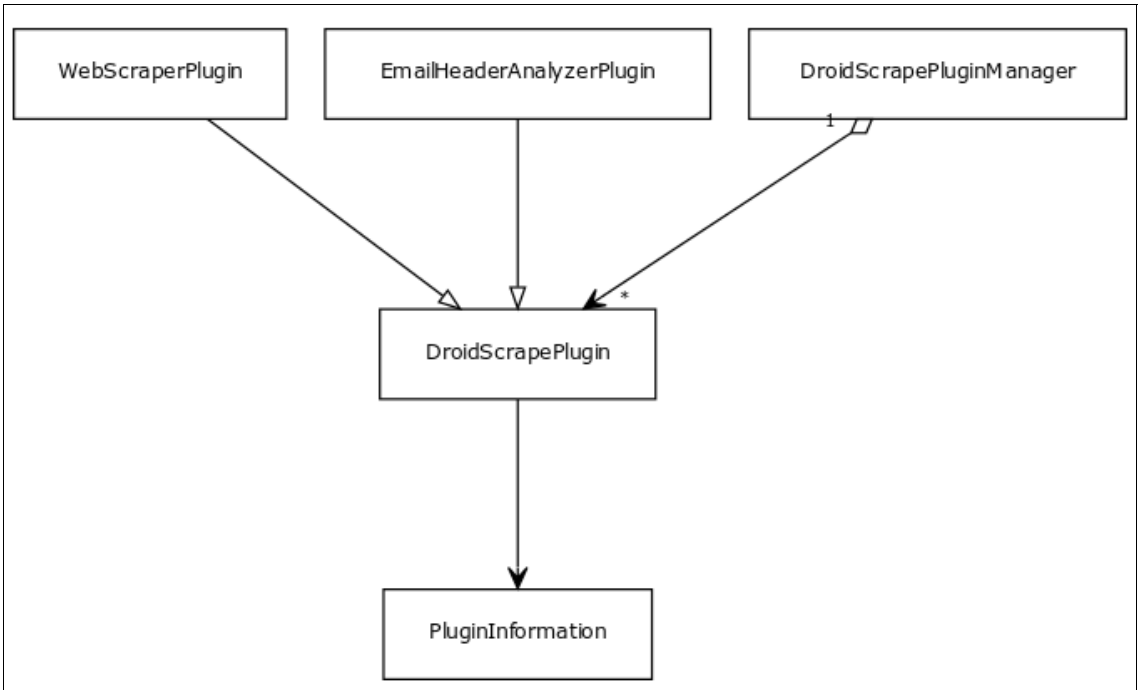


Ilustración 14: Diagrama de clases – Plugins

Cada uno de los plugins será el encargado de ejecutar sus propias actividades de forma independiente al flujo de la actividad principal. La actividad correspondiente a la pantalla de resultados dependerá de una actividad estandarizada para que los resultados de cada plugin sean compatibles con lo esperado por la aplicación. Para ello se utilizan clases que heredan de la interfaz de Android *Parcelable*[8] a través de la clase abstracta *DroidScrapeResultsReport*, que permite transmitir objetos complejos entre actividades.

Ilustración 15: Diagrama de clases - Plugin de análisis de cabeceras de email

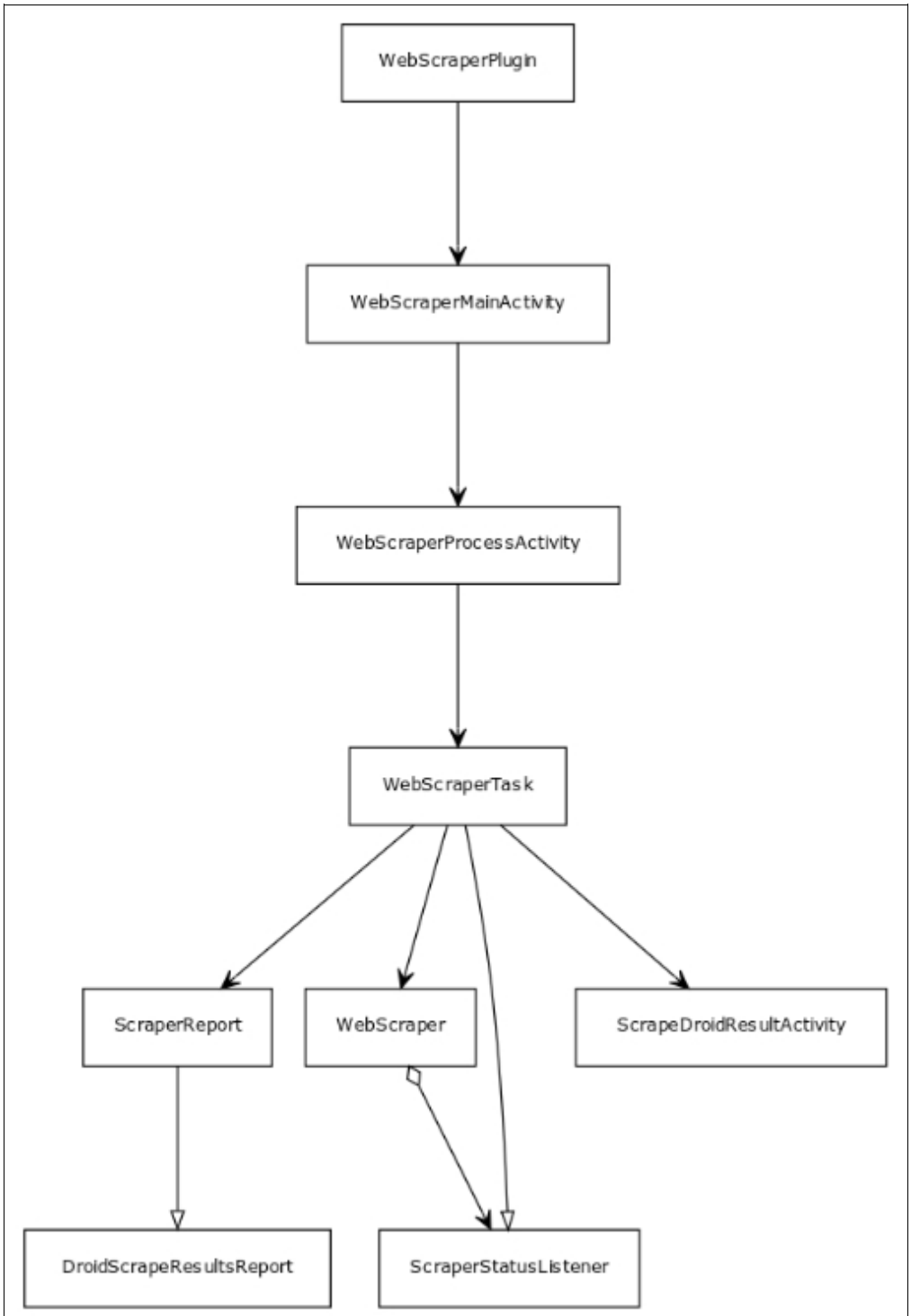


Ilustración 16: Diagrama de clases - Plugin de web scraping

Estos objetos “Empaquetables” son enviados a la actividad que mostrará los resultados.

3 Funcionamiento del Framework

En este capítulo se explicará el funcionamiento del framework a la hora de cargar los distintos plugins instalados.

3.1 Componentes de un plugin

Cada plugin deberá incluir, al menos, la información necesaria para que el framework pueda cargarlo y ejecutarlo. Esta información comprende los siguientes elementos:

- Un fichero JSON que contendrá toda la información sobre el plugin
- Una clase que extenderá a la clase abstracta DroidScrapePlugin
- Los componentes necesarios (actividades, iconos, etc) para ejecutar el plugin

El fichero JSON se deberá almacenar en la carpeta “raw” de los recursos de la aplicación y contendrá, al menos, la siguiente información con los siguientes campos:

- pluginName: Nombre del plugin
- pluginClass: Ruta completa de la clase principal del plugin
- pluginIcon: Nombre del fichero de icono del plugin, sin extensión

```
{
  "pluginName":"Email Header Analyzer",
  "pluginClass":"mdbdev.es.droidscrape.plugins.emailheaderanalyzer.EmailHeaderAnalyzerPlugin",
  "pluginIcon":"email"
}
```

Ilustración 17: Ejemplo de fichero de configuración de plugin

3.2 Flujo de actuación del Framework

1. El framework cargará toda la información de todos los ficheros de configuración de plugins presentes en la carpeta “raw” de los recursos de la aplicación, instanciando las clases principales especificadas en los mismos y acumulándolas en una lista.
2. Posteriormente, generará botones para cada uno de los plugins y los incluirá en el menú principal. A cada uno de los botones les asignará el método “start()” de la clase principal del plugin.
3. Al pulsar cada uno de los botones generados, se lanzará dicho método con las funcionalidades especificadas en la clase.

4 Pruebas unitarias

Se han incluido algunas pruebas unitarias básicas al proyecto que asegurarán el buen funcionamiento de algunos métodos básicos de los plugins implementados.

Dichas pruebas unitarias se pueden ejecutar desde la clase UnitTest (situada en la carpeta DroidScrape\app\src\test\java\mdbdev\es\droidscrape).

En ambos casos, debido a la característica modular de los plugins, se ha necesitado utilizar reflexión para poder ejecutar los métodos privados objeto de las pruebas.

4.1 Plugin de análisis de cabeceras de email

Para este plugin se han desarrollado tests unitarios que aseguren el correcto funcionamiento de los métodos encargados de las funciones nucleares de su actividad, cerciorándonos de que el análisis de cabeceras cumple los siguientes requisitos:

- El sistema identifica las direcciones IP privadas.
- El sistema identifica las direcciones IP remotas y obtiene datos al respecto. En este caso, la prueba depende de la disponibilidad de la API utilizada.
- El sistema es capaz de obtener información sobre dominios remotos.
- El sistema analiza todas las cabeceras presentes en el texto de entrada.
- El sistema publica actualizaciones para cada cabecera analizada a los *Listeners* configurados.
- Todas las cabeceras analizadas reciben un análisis no vacío ni nulo.

4.2 Plugin de scraper web

En el caso de este plugin se han desarrollado tests unitarios que prueban la funcionalidad de los métodos que se encargan de extraer enlaces de un documento HTML y buscar ocurrencias de términos en el mismo.

Debido a que no se se tiene acceso a un sitio web remoto completamente estático que admita abiertamente la actividad de crawlers y para no depender de la disponibilidad del mismo en caso de crearlo, no se ha podido desarrollar un test completo de la funcionalidad del plugin.

Las funcionalidades comprobadas en estos tests unitarios son:

- El sistema localiza todas las ocurrencias de los términos de búsqueda provistos y las incluye en los resultados.
- El sistema localiza todos los enlaces disponibles, de varios tipos, y los incluye en la lista de enlaces pendientes de visita.

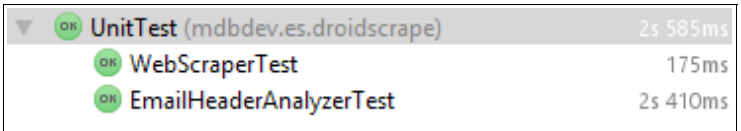


Ilustración 18: Resultados del test en la última versión de la aplicación

5 Resultados de la evaluación

El proceso de evaluación se ha dividido en tres fases cuyos resultados se explican a continuación.

5.1 Evaluación de alto nivel

Se proporcionaron los prototipos a los usuarios objeto de la evaluación, que propusieron los siguientes cambios:

- Inclusión de una barra de progreso en la pantalla de proceso del scraper web (Ilustración 10).
  - Se ha tenido en cuenta el cambio y se ha incluido en la primera versión funcional de la aplicación.
- Selección del nivel de profundidad del escaneo en el formulario de configuración del scraper web (Ilustración 9) mediante un spinner en lugar de una caja de texto, limitando así un escaneo excesivo que podría provocar una interrupción de la aplicación.
  - Se ha tenido en cuenta el cambio y se ha incluido en la primera versión funcional de la aplicación.

5.2 Evaluación del prototipo funcional

Gracias al buen cumplimiento de la planificación se ha podido llegar a esta fase con un prototipo funcional completo que cumple la gran mayoría de requisitos de la aplicación.

Se ha subido la aplicación a Google Play en fase alfa, de forma cerrada, y se ha dado acceso a varios usuarios potenciales de la aplicación.

En esta fase de evaluación se han obtenido, hasta el momento, dos posibles mejoras:

- Inclusión de un botón de *feedback* en la aplicación que permita a los usuarios proponer este tipo de cambios de forma más cómoda desde su dispositivo móvil.
- Mejora del tratamiento de entradas inválidas en el módulo de análisis de cabeceras de email.

Dichas mejoras se han llevado a cabo antes del inicio de la siguiente fase de evaluación.

5.3 Evaluación de la aplicación en beta

La aplicación ha sido publicada en Google Play en fase Beta. Se ha lanzado como beta abierta con un máximo de 1000 usuarios.

Se enviará a diversos usuarios potenciales un email invitándoles a probar la aplicación.

A lo largo de las próximas semanas se recogerá el feedback de los usuarios para acometer posibles mejoras.

6 Cambios en el diseño

En el proceso de desarrollo se han encontrado diferentes escollos que han provocado determinados cambios en el diseño de la aplicación. Por otra parte, gracias a las evaluaciones realizadas por usuarios potenciales, se han incluido algunos cambios sugeridos por los mismos.

En este apartado se explican los principales puntos en los que la aplicación, en su estado actual, difiere del diseño original.

6.1 Elementos eliminados

- Botón de “Copiar al portapapeles” en la pantalla de resultados: Se ha decidido eliminar este botón debido a que las limitaciones del portapapeles de Android no permitían copiar toda la información necesaria para darle utilidad. Por otra parte, la función de envío por email cubre las necesidades del usuario y, por ello, se considera que la utilización del portapapeles es secundaria.
- Opción de menú “Salir de la aplicación”: Se ha decidido no implementar esta opción debido a que su utilización podría causar la no eliminación de los archivos de informes de resultados generados por los distintos plugins. Por otra parte, contraviene el ciclo de vida propuesto por Android para sus Activities[6].

6.2 Elementos modificados

- Función “Enviar por email” de los informes de resultados: Debido a problemas de compatibilidad con versiones de Android anteriores a la 7.0, no se pueden generar correos electrónicos en formato HTML de forma directa. Los métodos necesarios para ejecutar este tipo de acciones han quedado obsoletos y podrían no recibir soporte en futuras versiones del sistema operativo[7]. Debido a esta circunstancia, se ha optado por generar un fichero y enviarlo por email como documento adjunto, lo que nos permite ofrecer un informe más vistoso al usuario sin renunciar a la comodidad de seleccionar su propio cliente de correo. Esta decisión tiene sin embargo una contrapartida negativa: obliga a generar un fichero que debe ser eliminado al cerrar la aplicación.
- Siguiendo las propuestas de cambio recibidas durante la fase de evaluación, se ha modificado el formulario del plugin de scraper web para que la selección de profundidad se realice mediante un control spinner limitado en lugar de un campo de texto. De esta forma se mejora el control de la aplicación sobre la entrada del usuario.

6.3 Elementos añadidos

- Opción “Forzar borrado de datos”: Como medida de seguridad, se ha incluido en el menú de la aplicación una opción que permite al usuario eliminar todos los datos generados por la aplicación.

- Pantalla de progreso del plugin de análisis de cabeceras de Email: Tras realizar distintas pruebas con la API utilizada para analizar los datos de las cabeceras se ha detectado que el proceso puede tardar varios segundos. Se ha decidido añadir una pantalla de progreso que transmita al usuario la impresión de que la aplicación está trabajando.
- Siguiendo las propuestas de cambio recibidas durante la fase de evaluación, se ha añadido una barra de progreso a la pantalla de progreso del plugin de scraper web, dando un aspecto visual más atractivo a dicha pantalla.

7 Conclusiones y agradecimientos

En primer lugar, me gustaría agradecer al equipo de la UOC y, especialmente, a Francesc D´Assís Giralt Queralt por su ayuda y buenos consejos que me han ayudado a completar este proyecto de la mejor manera posible. También a mis compañeros de trabajo, sin cuya ayuda no hubiera podido realizar la evaluación de la aplicación y que me han aportado ideas interesantes para la misma.

El proyecto ha resultado más manejable de lo que en un principio me planteé, gracias en parte a que la fase de planificación fue llevada a cabo con acierto. Esto denota la gran importancia que estas fases iniciales tienen para un proyecto de cierta magnitud.

La planificación inicial se ha podido cumplir en su totalidad, sin embargo, conseguirlo ha requerido ciertos esfuerzos debido a que, durante algunos periodos de tiempo, la carga de trabajo ajena al proyecto ha sido excesiva y no ha permitido cumplir con los horarios planeados. En cualquier caso, se han podido cumplir todos los hitos hasta el momento en tiempo y forma.

La evaluación de la aplicación ha dado frutos, habiendo aportado los usuarios feedback valioso que ha servido para mejorar notablemente la aplicación. En muchos casos, los desarrolladores tenemos la tendencia a acometer proyectos sin tener en cuenta al usuario objetivo hasta el momento en el que los publicamos. Si bien esto no tiene por qué afectar a la calidad de la lógica, sí afecta claramente a la experiencia del usuario y puede por tanto arruinar proyectos que han supuesto una gran inversión de tiempo e incluso de dinero.

Por tanto, la principal conclusión extraida del trabajo realizado es que las fases iniciales de planteamiento del proyecto pueden ser tan importantes como el desarrollo y, cuanto mayor es la envergadura del proyecto, más relevancia tienen.

8 Próximos pasos

En este apartado se perfilan algunas de las ideas que posiblemente se implementen en un futuro con el objetivo de ampliar la funcionalidad de la aplicación o mejorar su calidad.

8.1 Desarrollo de nuevos plugins

Durante las primeras fases del proyecto se idearon algunos plugins que, por diversas razones, no fueron incluidos en el alcance inicial.

- Plugin de RSS temático: Un agregador RSS que muestre las noticias o investigaciones más importantes relacionadas con el mundo de la ciberseguridad y la ciberinteligencia.
- Plugin de comprobación de listas negras: Un plugin que permita consultar si un dominio o una IP está presente en las principales listas negras de spam, malware, etc.

8.2 Mejoras estéticas de la interfaz

Para hacer que la aplicación sea más atractiva, se creará un nuevo tema visual con colores y *layout* más atractivos.

8.3 Mejoras en el plugin de análisis de cabeceras

Utilizando las APIs de Google Maps, se generarán mapas que muestren los dominios y direcciones IP geolocalizados detectados por la aplicación.

Estos mapas se incluirán en el informe generado por el plugin.

8.4 Mejoras propuestas por los usuarios

La aplicación está publicada actualmente en fase beta en Google Play. El feedback recogido durante esta fase será empleado para acometer mejoras en la aplicación.

9 Glosario

Término	Significado
OSINT	Open Source Intelligence, Inteligencia en fuentes abiertas.
MVC	Patrón de diseño Modelo-Vista-Controlador
Web Scraping	Técnica de extracción de información de un sitio web de forma automatizada, simulando la navegación de un humano.

10 Bibliografía

1. Matt Asay, «Moore's Law Is Dead! (But Not In Mobile)» http://readwrite.com/2015/04/20/revenge-of-moores-law-on-mobile/ 20/04/2015
2. Alyssa Newcomb, «Intel CEO Says Reports of the Death of Moore's Law Have Been Greatly Exaggerated» http://abcnews.go.com/Technology/intel-ceo-reports-death-moores-law-greatly-exaggerated/story?id=38703042 27/04/2016
3. Free Software Foundation, «GNU General Public License» https://www.gnu.org/licenses/gpl-3.0.en.html 08/03/2017
4. Net Market Share, «Mobile/Tablet Operating System Market Share» https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1 08/03/2017
5. Android Developer Portal https://developer.android.com/about/dashboards/index.html 08/03/2017
6. Android Developer Portal https://developer.android.com/guide/components/activities/activity-lifecycle.html 03/05/2017
7. Android Developer Portal https://developer.android.com/reference/android/text/Html.html 03/05/2017
8. Android Developer Portal https://developer.android.com/reference/android/os/Parcelable.html 22/05/2017

11 Anexos

11.1 Anexo I: Análisis de herramientas similares

Scrapy
https://scrapy.org/

Scrapy es un framework escrito en Python que permite la creación de forma cómoda de arañas o scrapers web.

Una de las grandes ventajas de Scrapy es su extensibilidad y la amplia comunidad que soporta esta herramienta.

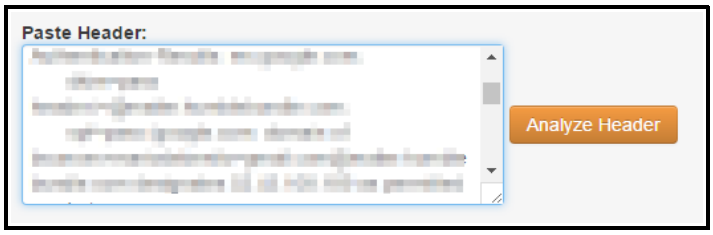
Además, sus creadores ofrecen un servicio en la nube para alojar los scrapers desarrollados por los usuarios y poder tener acceso a ellos desde cualquier lugar.

Su principal desventaja es la necesidad de tener capacidades de programación y conocer el lenguaje Python para poder utilizarlo, ya que los scrapers se deben desarrollar.

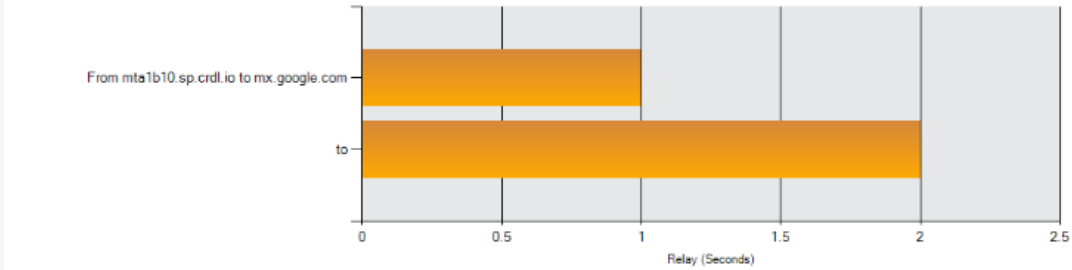
MxToolbox
https://mxtoolbox.com

MxToolbox es un sitio web especializado en el análisis de inteligencia relacionado con el correo electrónico.

Entre sus servicios destaca el analizador de cabeceras de correo, que extrae toda la información disponible en dichas cabeceras y la muestra de forma sencilla e intuitiva.



En este analizador, basta con introducir la cabecera en texto plano para poder obtener un completo informe que desgrana toda la información contenida en la misma así como un listado de los saltos que el correo ha realizado para llegar a su destino.



Hop	Delay	From	By	With	Time (UTC)	Blacklist
1	*	mta1b10.sp.crdl.io 52.43.108.189	mx.google.com	ESMTPS	3/6/2017 7:29:36 PM	
2	1 Second		10.83.11.17	SMTP	3/6/2017 7:29:37 PM	

Por otra parte, este servicio mantiene una serie de listas negras que permiten identificar rápidamente si un correo ha pasado por algún servidor sospechoso.

IP2Location email tracer
http://www.ip2location.com/free/email-tracer

Este servicio es capaz de definir los distintos servidores por los que un correo electrónico ha pasado antes de llegar a su destino. Al introducir la cabecera del correo a analizar, muestra una lista detallada de la localización de las direcciones IP de todos los saltos que ha realizado el correo.

11.2 Anexo II: API de geolocalización

Para el desarrollo de esta aplicación se ha utilizado la API de geolocalización de IP-API.com (http://ip-api.com/docs/).

Esta API permite, de forma gratuita y con un amplio límite, geolocalizar las direcciones IP encontradas en las cabeceras de email analizadas.

11.3 Anexo III: Manual de usuario

En este anexo se incluye un manual de usuario de la aplicación detallando el modo de uso de sus principales funciones.

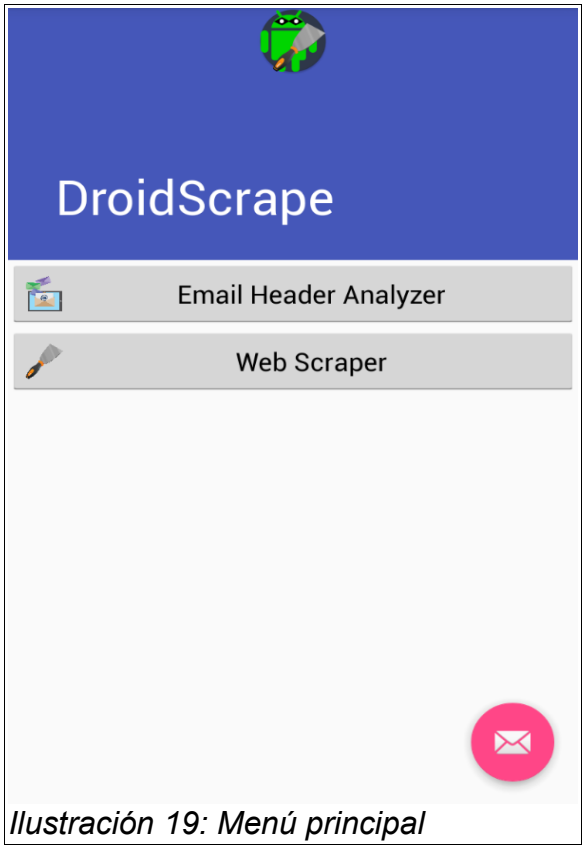
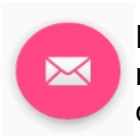


Ilustración 19: Menú principal

Feedback y sugerencias



En el menú principal se puede ver un botón flotante con un icono representando un sobre que permite remitir sugerencias o feedback directamente al desarrollador.

Módulo de Análisis de Cabeceras de Email

Para realizar el análisis de un texto que contenga cabeceras de email bastará con pegarlo en la pantalla principal del plugin y pulsar en el botón “Analyze”.

La aplicación procesará los datos de la cabecera durante un periodo de tiempo que dependerá de la disponibilidad de las APIs de terceros utilizadas.

Una vez concluido este proceso, mostrará un informe de resultados con la resolución de todos los dominios y direcciones IP presentes en las cabeceras.

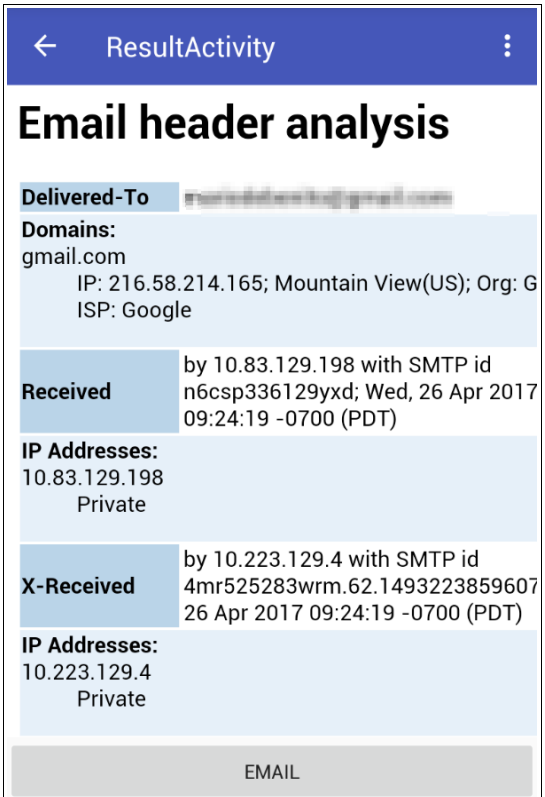


Ilustración 20: Analizador de cabeceras - informe de resultados

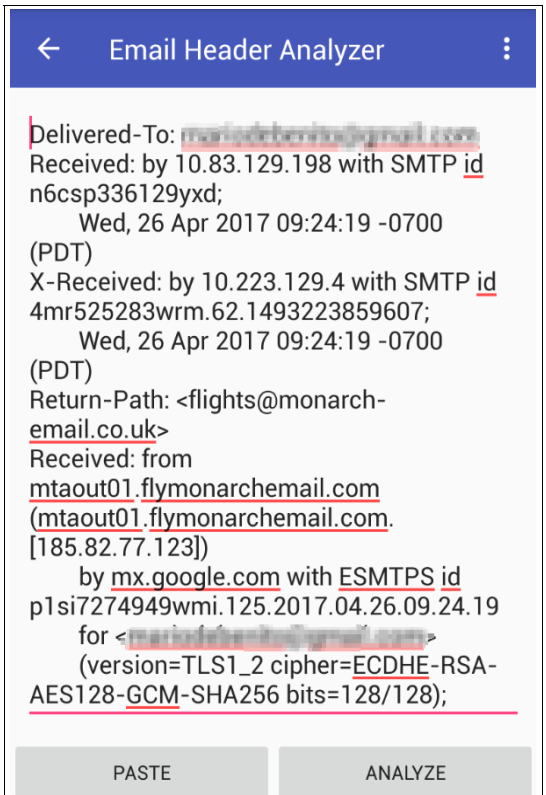


Ilustración 21: Analizador de cabeceras - pantalla inicial

Este informe se podrá enviar cómodamente por email pulsando en el botón “Email” situado en la parte inferior de la pantalla.

Módulo de Scraper Web

Para realizar un proceso de *crawling* sobre un sitio web simplemente se deberán configurar las opciones de la pantalla principal del plugin para después pulsar en el botón “Start!”.

El formulario se puede limpiar fácilmente pulsando en el botón “Clear”.

Las distintas opciones disponibles son las siguientes:

- URL a *crawlear*: Se deberá introducir la URL principal del sitio web que se desea *crawlear*.
- Nivel de profundidad de escaneo: Con opciones desde 0 hasta 3, esta selección marca el nivel de profundidad de enlaces que la aplicación escaneará. El nivel de profundidad indica cuantos enlaces anidados indexará y revisará el proceso:
  - 0: Únicamente escaneará la página principal.
  - 1: Escaneará la página principal y todas las páginas enlazadas en ella.

    - 2, 3: Escaneará todas las páginas anteriores y todos las que se encuentren enlazadas en ellas.
- Modo de escaneo: Definirá la agresividad con la que se comportará el *crawler*. Un proceso de *crawling* demasiado agresivo puede provocar que el sitio web objetivo tome medidas defensivas pudiendo llegar incluso a banear el dispositivo del usuario.
    - Soft: Realizará pausas largas entre cada petición web.
    - Normal: Realizará pausas cortas entre cada petición web.
    - Aggressive: Realizará pausas mínimas entre cada petición.
- Términos de búsqueda: Cadenas de texto que se buscarán en el sitio web escaneado. Se deberán introducir una en cada línea.

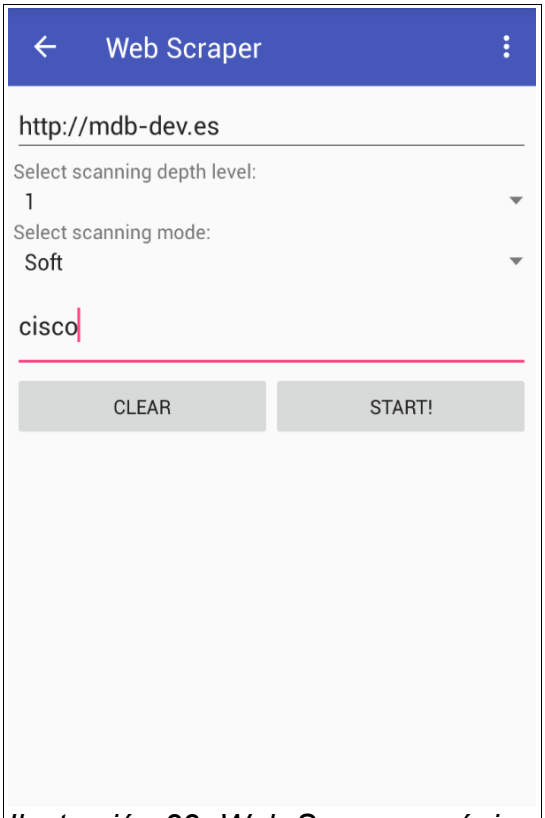


Ilustración 22: Web Scraper - página de configuración

Una vez iniciado el escaneo, el progreso del mismo se mostrará en la pantalla. En el primer campo, se indicará si el *crawler* está trabajando o esperando. En el segundo, se podrá ver el progreso actual del escaneo. También se nos indica la URL que está siendo analizada actualmente y la última que se analizó por completo.

El escaneo se puede detener en cualquier momento pulsando en el botón “Cancel”.

Ilustración 23: Web Scraper - escaneo en progreso

Ilustración 24: Web Scraper - resultados del escaneo

Una vez finalizado el escaneo, se mostrará el informe de resultados indicando todas las apariciones de cada término junto a su contexto HTML.

Este informe se puede enviar fácilmente por email pulsando en el botón “Email”.

Limpieza de ficheros

Para preservar la confidencialidad de los datos analizados los ficheros generados por la aplicación para su envío por email son eliminados automáticamente al salir de la misma.

Para mayor seguridad, se puede forzar su borrado en cualquier momento desde el menú superior mediante la opción “Clear files”.

NOTA: No es conveniente eliminar los ficheros inmediatamente después de enviarlos por email. Algunos clientes de correo pueden tardar algunos minutos en procesar los ficheros adjuntos y, si estos son eliminados antes, el envío podría quedar incompleto.

11.4 Anexo IV: Manual del desarrollador

En este anexo se detallarán algunos conceptos que permitirán la ampliación de la aplicación mediante nuevos plugins.

Dependencias de la aplicación

La aplicación se ha desarrollado utilizando las siguientes librerías de terceros:

- Google GSON (2.7): Librería de serialización / deserialización de objetos Java en formato JSON.

  https://github.com/google/gson

- Jsoup (1.8.3): Librería que simplifica las labores de *parsing* de ficheros HTML y realización de peticiones HTTP.

  https://jsoup.org/

Construcción de un plugin

La aplicación cargará y permitirá la ejecución de forma automática de los plugins desarrollados según las directrices indicadas a continuación.

Para construir un nuevo plugin se deberán generar los siguientes contenidos:

- Fichero JSON con la definición del plugin. Un ejemplo de este fichero se puede ver en la Ilustración 17. Deberá ser colocado en la carpeta “raw” del directorio de recursos de la aplicación. Este fichero deberá contener la siguiente información:
  - pluginName: Nombre del plugin
  - pluginClass: Ruta completa de la clase principal del plugin
  - pluginIcon: Nombre del fichero de icono del plugin, sin extensión
- Imagen con el icono del plugin. Deberá ser colocada en la carpeta “drawable” del directorio de recursos del proyecto. Su nombre de archivo, sin incluir la extensión, deberá coincidir con el especificado en el fichero JSON.
- Clase principal del plugin: Deberá extender a la clase “DroidScrapePlugin”. La ruta completa a esta clase deberá ser la especificada en el fichero JSON.
- Actividades propias del plugin: El plugin podrá ejecutar cualquier tipo de actividades de forma independiente a la aplicación aún estando contenida en esta.
- Dependencias del plugin: En caso de que el plugin requiera de alguna librería de terceros para funcionar deberá ser incluida en el fichero “build.gradle” de la aplicación.

La clase principal del plugin deberá implementar el método “start()” de la clase “DroidScrapePlugin”. En este método podrá ejecutar el código necesario para lanzar cualquier proceso o actividad.

En caso de que se desee utilizar la Actividad estándar para mostrar resultados, “DroidScrapeResultActivity”, Se puede extender la misma o emplearla directamente. La actividad necesita recibir un objeto de tipo “DroidScrapeResultsReport” para mostrar el informe HTML y generar el archivo correspondiente para su envío por correo electrónico.

11.5 Anexo V: Publicación de la aplicación

La aplicación está disponible actualmente en fase Beta en Google Play:
https://play.google.com/store/apps/details?id=mdbdev.es.droidscrape

El código fuente de la misma se puede encontrar en el siguiente repositorio:
https://gitlab.com/mdebenito/DroidScrape