



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea



Escuela Universitaria
de Ingeniería
Vitoria-Gasteiz

Ingeniariartzako
Unibertsitate Eskola
Vitoria-Gasteiz

Desarrollo de herramientas de seguridad informática para Android

Memoria del Trabajo de Fin de Grado

presentada para optar al grado de

Ingeniería Informática de Gestión y Sistemas de Información

por

Ander Granado Masid

Director: Pablo González Nalda

18 de abril de 2017

Agradecimientos

Índice general

Agradecimientos	III
Resumen	XIII
I. Alcance del Trabajo	1
1. Descripción, Objetivos y Motivación del proyecto	3
1.1. Descripción	3
1.2. Objetivos	4
1.3. Motivación	4
2. Viabilidad	5
2.1. Requisitos funcionales del trabajo	5
2.2. Planificación del tiempo	6
2.2.1. EDT	6
2.2.1.1. Fase 1	8
2.2.1.2. Fase 2	9
2.2.2. Agenda del proyecto	9
2.2.3. Tareas	10
2.2.4. Entregables	20
2.2.4.1. Fase 1	20

2.2.4.2. Fase 2	20
2.2.5. Cronograma	21
2.3. Gestión de costos	24
2.3.1. Presupuesto	25
2.4. Gestión de riesgos	28
2.4.1. Explicación y plan de contingencia	28
 II. Fase 1: Estado del Arte de la Seguridad Informática	 33
 3. Conceptos Generales	 35
3.1. Seguridad Informática	36
3.2. Seguridad de la Información	36
3.2.1. Diferencias respecto a la Seguridad Informática	36
3.3. Servicios de la Seguridad de la Información	36
3.3.1. CID	36
3.3.2. Otros servicios	36
3.3.2.1. Autenticación	36
3.3.2.2. Anonimato	36
3.3.2.3. Protección a la réplica	36
 4. Aplicaciones	 37
 5. Pentesting	 39
5.1. Objetivos	40
5.2. Partes	40
5.2.1. Recogida de información	40
5.2.1.1. Internal Footprinting	40
5.2.1.2. External Footprinting	40
5.2.2. Análisis de vulnerabilidades	40
5.2.3. Ataques de penetración	40
5.2.3.1. Ataques de contraseñas	40
5.2.3.2. Exploits	40
5.2.3.3. Auditoría de Aplicaciones Web	40
5.2.3.4. Ataques a redes	40

6. Conclusiones	41
III. Fase 2: Desarrollo de la aplicación	43
7. Introducción	45
8. Tecnologías y herramientas	47
8.1. Kali Linux	47
8.2. NMap	47
8.3. Android	47
8.3.1. Android SDK	47
8.3.2. Android Studio	47
8.4. Otras herramientas	47
8.4.1. Git	47
9. Desarrollo de la aplicación	49
10. Testeo y corrección de errores	51
IV. Análisis y conclusiones del Trabajo	53
V. Apéndices	55

Índice de figuras

2.1.	EDT completo	7
2.2.	EDT de la Fase 1	8
2.3.	EDT de la Fase 2	9
2.4.	Cronograma de la fase inicial	21
2.5.	Cronograma de la Fase 1	22
2.6.	Cronograma de la Fase 2	23
2.7.	Cronograma de la elaboración de la memoria	24
2.8.	Recursos de trabajo y materiales	24

Índice de cuadros

2.1. Calendario de días festivos oficial (de Álava)	10
2.2. Reducciones de la jornada laboral	10
2.3. Recursos materiales (software)	25
2.4. Recursos de trabajo	25
2.5. Recursos materiales (hardware)	26
2.6. Recursos materiales (software)	26
2.7. Costo de recursos de trabajo	26
2.8. Costo de recursos materiales	26
2.9. Amortizaciones de hardware y de software	27
2.10. Total presupuesto	27
2.11. Enumeración de riesgos del proyecto	28

Resumen y Organización de la memoria

I

Alcance del Trabajo

Descripción, Objetivos y Motivación del proyecto

1.1. Descripción

En este trabajo se desarrolla un estudio sobre el campo de la seguridad informática, mas concretamente sobre las diferentes herramientas de seguridad informática. En base a eso, se desarrolla una aplicación para dispositivos móviles que busca, de manera sencilla para el usuario, proporcionar soluciones a tareas recurrentes dentro del campo de la seguridad informática, basandose en herramientas ya existentes. Estas herramientas, usadas por pentesters, analistas forenses o hackers de sombrero blanco permiten elaborar operaciones de todo tipo, desde escanear una red inalámbrica hasta romper el cifrado de un archivo para acceder a la información que contiene.

Gran parte de estas herramientas son gratuitas [Oficina de Seguridad del Internauta(2017)] o incluso de software libre [GitHub, Inc(2017)], lo que otorga la posibilidad de que dichas herramientas mejoren continuamente.

Sin embargo, el mayor problema de este tipo de herramientas suelen ser su público objetivo. Normalmente este tipo de herramientas están diseñadas para profesionales del sector, profesionales tanto con conocimientos de seguridad informática como de programación o administración de sistemas. La mayoría de estas herramientas se basan en librerías o frameworks completos, con cierta dificultad de uso, o scripts CLI. Debido a esto, cierta tarea como escanear una red, que para un experto en ciberseguridad o un administrador de sistemas se convierte en 5 segundos tecleando un comando, para un usuario medio se convierte en un auténtico quebradero de cabeza.

1.2. Objetivos

El objetivo de este proyecto es doble. Por una parte se busca realizar un análisis del campo de la seguridad informática, un *estado del arte* del área que permita vislumbrar cuales son las diferentes aplicaciones de dicho área y a partir de ahí concretar las necesidades más importantes dentro de ese campo para, al final, acabar elaborando una aplicación para Android que nos proporcione ciertas utilidades.

Por otra parte, también se busca que la aplicación a elaborar sirva tanto para usuarios experimentados en la materia como para un público general. Para ello, un buen diseño de la interfaz gráfica (GUI) o diferentes principios de experiencia de usuario (UX) jugarán un papel fundamental. De esta manera lograremos una transición entre herramientas accesibles solo para unos pocos a una herramienta para todo el mundo.

1.3. Motivación

A día de hoy la informática es un industria fundamental dentro de la sociedad en general y de las vidas de las personas en particular. Los ordenadores personales son la herramienta fundamental de trabajo en una gran cantidad de áreas, además de una herramienta que se encuentra en prácticamente cualquier hogar. Los denominados Smartphones junto a Internet se han convertido en la principal herramienta de comunicación. La revolución causada por la industria llega hasta tal punto que una organización como la ONU ha declarado Internet *como un derecho humano por ser una herramienta que favorece el crecimiento y el progreso de la sociedad en su conjunto* [El Mundo(2011)],.

Teniendo en cuenta toda la información que transmitimos, almacenamos y procesamos, sería lógico pensar que la seguridad de dicha información es vital. El área de la seguridad informática se encarga de ofrecer los mecanismos necesarios para que nuestra información no se vea comprometida de ninguna manera y nuestros dispositivos permanezcan seguros y con la menor cantidad de vulnerabilidades posible. Cada vez este área resulta mas importante, y sobre todo ahora con la llegada del Internet of Things (IoT), ya que pasamos de tener no solo nuestros ordenadores o Smartphones conectados a Internet, sino que tenemos otros dispositivos como nuestro coche o nuestra lavadora conectados, con los riesgos que ello conlleva.

Este capítulo tiene como objetivo realizar, en base a los objetivos marcados en el capítulo anterior, elaborar un análisis de viabilidad del proyecto, analizando las diferentes tareas que contiene, calculando los correspondientes gastos y sus inherentes riesgos, que pueden afectar, atrasando o incluso impidiendo, la realización del proyecto.

2.1. Requisitos funcionales del trabajo

Los requisitos funcionales de la aplicación (RF a partir de ahora) se elaboran en base a los objetivos descritos y son los siguientes:

- Elaborar un estado del arte que analice el campo de la seguridad informática, sus aplicaciones y analice el área del pentesting
- Elaborar una aplicación que permita obtener información sobre redes y nodos de la red
- Integrar correctamente herramientas de terceros para lograr que la aplicación resulte lo mas escalable posible.
- Aplicar principios sobre experiencia de usuario (UX) y sobre el diseño de interfaces gráficas (GUI) para que la aplicación sea lo mas sencilla y cómoda de usar

2.2. Planificación del tiempo

2.2.1. Estructura de Descomposición del Trabajo

El EDT, o Estructura del Desglose del Trabajo, es un sistema jerárquico que permite organizar las diferentes tareas de un proyecto. Es una técnica ampliamente usada para gestionar todo tipo de proyectos, especialmente proyectos de software.

A la hora de planificar el tiempo, se ha tenido en cuenta un enfoque en dos fases, a las cuales denominaremos Fase 1 y Fase 2. Esto es debido al carácter del proyecto. Por una parte, para elaborar la aplicación anteriormente mencionada, resulta fundamental realizar un estudio sobre el campo de la seguridad informática, mas concretamente sobre el área del pentesting, para poder llegar a encontrar las mejores herramientas y técnicas que permitan desarrollarla. Este estudio, bien desglosado y fundamentado, llevara a la obtención de un elaborado estado del arte, que sera el principal objetivo de la Fase 1. Además, dicha fase contiene un periodo de aprendizaje y familiarización con diversos conceptos y tecnologías, que también quedaran reflejados.

En la Fase 2, en función de lo aprendido en la Fase 1, se elaborará la aplicación en base a los criterios de implementar diversas utilidades junto a un experiencia de usuario (UX) óptima, que vendrá acompañada de un buen diseño de una interfaz gráfica (GUI).

El EDT completo quedaría tal y como se muestra en la figura 2.1.

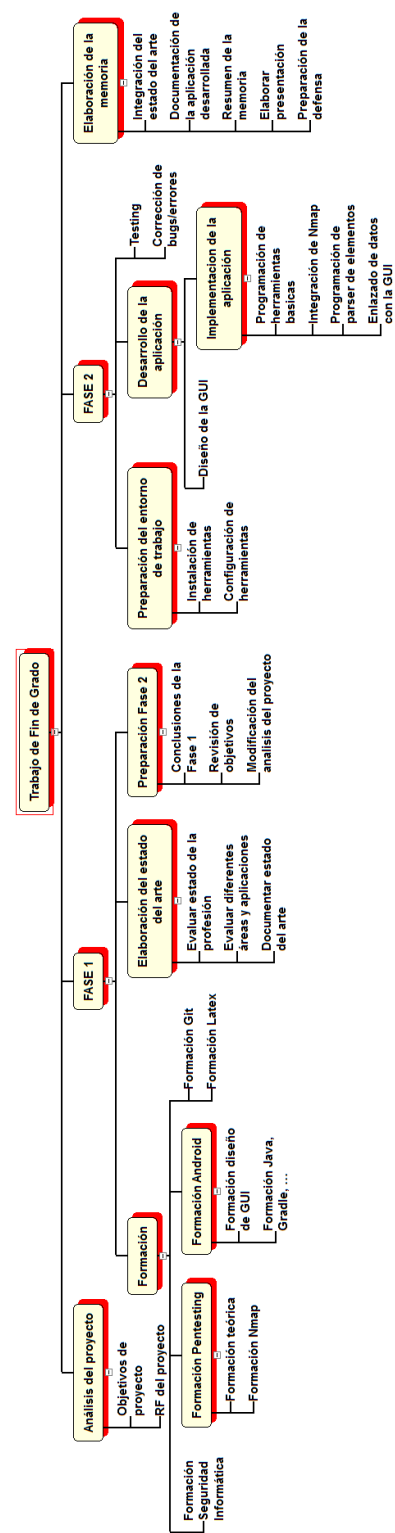


Figura 2.1.: EDT completo

2.2.1.1. Fase 1

El EDT para la Fase 1 quedaría como se muestra en la figura 2.2..

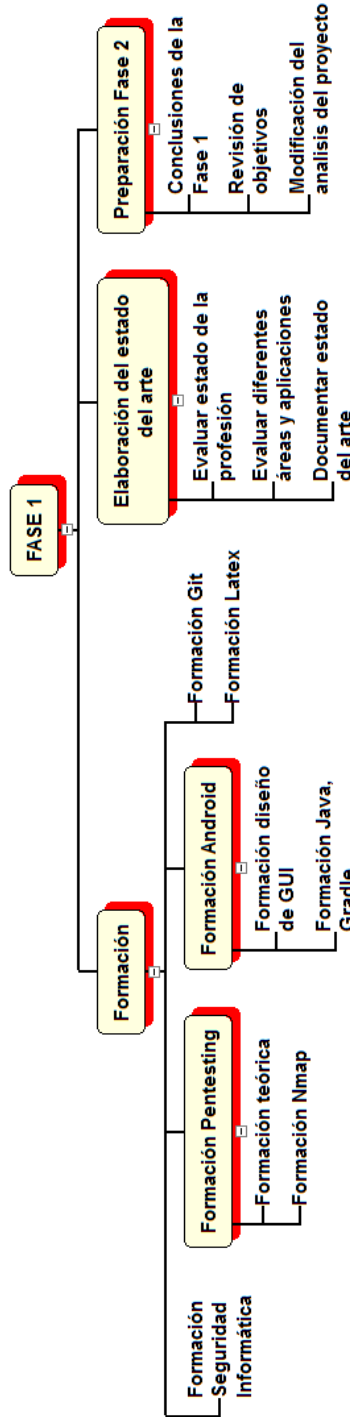


Figura 2.2.: EDT de la Fase 1

2.2.1.2. Fase 2

El EDT para la Fase 2 quedaría como se muestra en las figura 2.3.

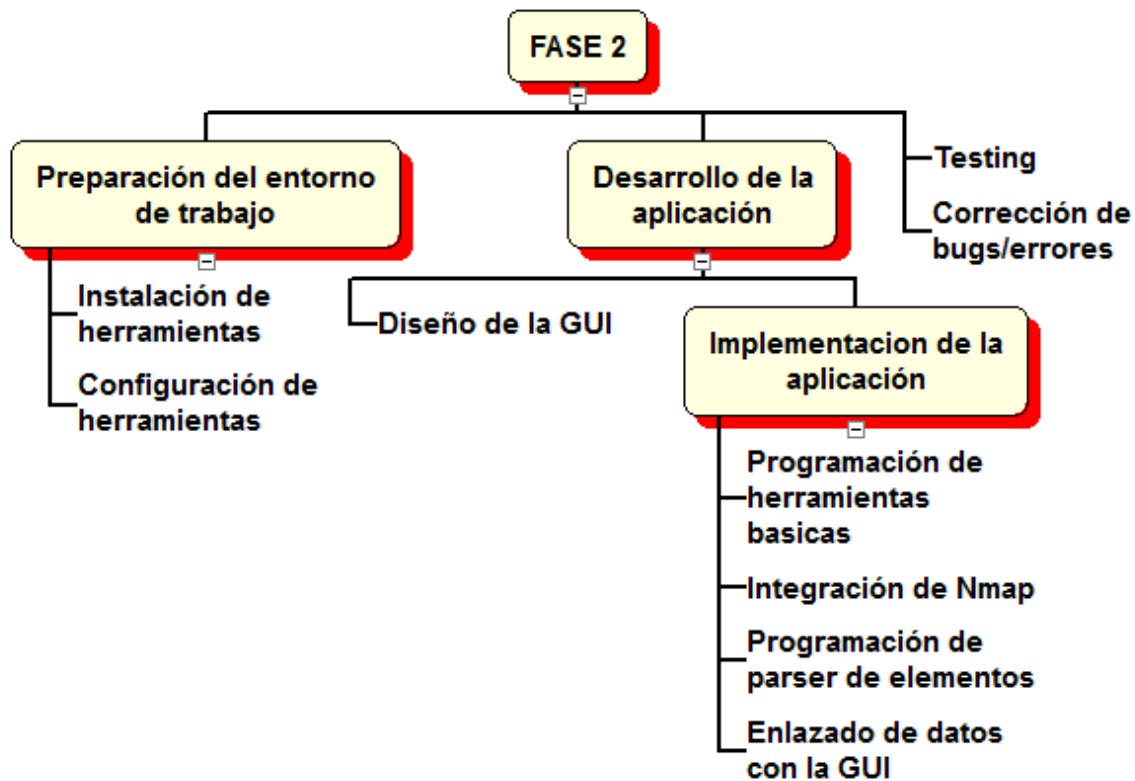


Figura 2.3.: EDT de la Fase 2

2.2.2. Agenda del proyecto

El proyecto se llevara a cabo durante varios meses, comenzando en abril. Se trabajará a media jornada (4 horas) de lunes a viernes, con las siguientes excepciones. Primero, se tendrá en cuenta el calendario de festivos oficiales para aplicar algunas jornadas festivas en cuyos días no se trabajará. Esos días quedan reflejados en la tabla 2.1. A esos festivos se les aplica unas excepciones, en las cuales, aun siendo festivo, se trabajará igualmente. Esos festivos en los que se trabaja están indicados en rojo.

Fecha	Evento
1 de Enero	Año nuevo
6 de Enero	Día de Reyes
19 de Marzo	San José
2 de Abril	Jueves Santo
3 de Abril	Viernes Santo
6 de Abril	Lunes de Pascua
28 de Abril	San Prudencio
1 de Mayo	Día del trabajo
25 de Julio	Santiago Apóstol
5 de Agosto	Virgen Blanca
15 de Agosto	Asunción de la Virgen
12 de Octubre	Fiesta Nacional de España
8 de Diciembre	Inmaculada Concepción
25 de Diciembre	Navidad

Cuadro 2.1.: Calendario de días festivos oficial (de Álava)

Por otra parte, existen una serie de días en los que se aplica una reducción de la jornada laboral de 4 a 2 horas (en los días considerados laborales), debido básicamente a exámenes. Estos periodos vienen indicados en la tabla 2.2.

Nombre del periodo	Fecha de inicio	Fecha de fin
Exámenes Convocatoria Ordinaria	16 de mayo	23 de mayo
Exámenes Convocatoria Extraordinaria	27 de junio	4 de julio

Cuadro 2.2.: Reducciones de la jornada laboral

En base a dicho calendario, la fecha estimada de finalización del proyecto es del 24 de Julio.

2.2.3. Tareas

El EDT completo de tareas, al que se añaden la fase inicial de objetivos del proyecto, y toda la fase final de elaboración de la memoria, presentación y la defensa quedaría de la siguiente

manera.

0. Análisis del proyecto

0.1 Objetivos del proyecto

0.2 RF del proyecto

1. FASE 1

1.1 Formación

1.1.1 Formación seguridad informática

1.1.2 Formación Pentesting

1.1.2.1 Formación Teórica

1.1.2.2 Formación NMap

1.1.3 Formación Android

1.1.3.1 Formación Diseño de GUI

1.1.3.2 Formación Java, Gradle, ...

1.1.4 Formación Git

1.1.5 Formación L^AT_EX

1.2 Elaboración del estado del arte

1.2.1 Evaluar estado de la profesión

1.2.2 Evaluar diferentes áreas y aplicaciones

1.2.3 Documentar estado del arte

1.3 Preparación de la Fase 2

1.3.1 Conclusiones de la Fase 1

1.3.2 Revisión de objetivos

1.3.3 Modificación del análisis del proyecto

2. FASE 2

2.1 Preparación del entorno de trabajo

2.1.1 Instalación de herramientas

2.1.2 Configuración de herramientas

2.2 Desarrollo de la aplicación

2.2.1 Diseño de la GUI

2.2.2 Implementación de la aplicación

2.2.2.1 Programación de herramientas básicas

2.2.2.2 Integración de NMap

2.2.2.3 Programación de parser de elementos

2.2.2.4 Enlazado de datos con la GUI

2.2.3 Testeo

2.2.4 Corrección de bugs/errores

3. Elaboración de la memoria

3.1 Integración del estado del arte

3.2 Documentación de la aplicación desarrollada

3.3 Resumen de la memoria

3.4 Elaborar presentación

3.5 Preparación de la defensa

4. Reuniones periódicas

A continuación se explica mediante una breve definición en que consiste cada tarea, además de especificar su duración en horas.

Número: 0.1.

Nombre: Objetivos del proyecto.

Descripción: Definir los objetivos que tiene que cumplir el TFG.

Trabajo estimado: 5 horas.

Número: 0.2.

Nombre: RF del proyecto.

Descripción: Definir, en base a los objetivos del proyecto, los Requisitos funcionales concretos del proyecto.

Trabajo estimado: 5 horas.

Número: 1.1.1.

Nombre: Formación seguridad informática.

Descripción: Familiarizarse con el amplio entorno de la seguridad informática y comprender las diferentes áreas, objetivos y el estado de dicho campo.

Trabajo estimado: 30 horas.

Número: 1.1.2.1.

Nombre: Formación Teórica.

Descripción: Familiarizarse con los conceptos de Pentesting, las diferentes técnicas usadas y las diferentes fases del proceso de Pentesting.

Trabajo estimado: 20 horas.

Número: 1.1.2.2.

Nombre: Formación NMap.

Descripción: Familiarizarse con el entorno de NMap, como implementarlo, usarlo para obtener información y de que formas se puede obtener información estructurada y organizada para su posterior uso.

Trabajo estimado: 10 horas.

Número: 1.1.3.1.

Nombre: Formación Diseño de GUI.

Descripción: Aprender a usar herramientas de diseño de GUI, diferentes patrones de Diseño en sistemas Android, y el uso de IDEs o herramientas para desarrollar dichas GUIs.

Trabajo estimado: 10 horas.

Número: 1.1.3.2.

Nombre: Formación Java, Gradle,

Descripción: Aprender sobre el uso de Java para desarrollar aplicaciones Android, diferentes clases, utilidades o conceptos recurrentes en la programación para Android.

Trabajo estimado: 15 horas.

Número: 1.1.4.

Nombre: Formación Git.

Descripción: Aprender el uso de dicho sistema de control de versiones para llevar un control riguroso del desarrollo del proyecto y de la aplicación.

Trabajo estimado: 5 horas.

Número: 1.1.5.

Nombre: Formación L^AT_EX.

Descripción: Aprender diferentes conceptos de L^AT_EX para elaborar tanto el estado del arte como el propio informe de la manera mas clara y elegante posible.

Trabajo estimado: 5 horas.

Número: 1.2.1.

Nombre: Evaluar estado de la profesión.

Descripción: Analizar los diferentes campos de la profesión, las necesidades mas demandadas y los diferentes perfiles de profesionales dentro del campo.

Trabajo estimado: 5 horas.

Número: 1.2.2.

Nombre: Evaluar diferentes áreas y aplicaciones.

Descripción: Evaluar las necesidades concretas a nivel técnico, las aplicaciones mas usadas y las virtudes y carencias de éstas.

Trabajo estimado: 10 horas.

Número: 1.2.3.

Nombre: Documentar estado del arte.

Descripción: Elaborar la documentación en base a toda la información recogida para obtener un elaborado estado del arte.

Trabajo estimado: 5 horas.

Número: 1.3.1.

Nombre: Conclusiones de la Fase 1.

Descripción: Elaborar una serie de conclusiones en función a todo el estudio realizado sobre el campo de la seguridad informática.

Trabajo estimado: 5 horas.

Número: 1.3.2.

Nombre: Revisión de objetivos.

Descripción: Revisión de los objetivos y los requisitos funcionales de la aplicación a desarrollar en función a todo lo investigado.

Trabajo estimado: 5 horas.

Número: 1.3.3.

Nombre: Modificación del análisis del proyecto.

Descripción: Modificar la parte de análisis del proyecto realizada anteriormente, antes de comenzar con la Fase 1.

Trabajo estimado: 5 horas.

Número: 2.1.1.

Nombre: Instalación de herramientas.

Descripción: Instalación de todo lo necesario para desarrollar la aplicación.

Trabajo estimado: 5 horas.

Número: 2.1.2.

Nombre: Configuración de herramientas.

Descripción: Configuración de todas las herramientas para que el desarrollo de la aplicación sea lo mas cómodo posible.

Trabajo estimado: 5 horas.

Número: 2.2.1.

Nombre: Diseño de la GUI.

Descripción: Diseñar una interfaz gráfica clara y sencilla de usar para interactuar con las funciones a implementar.

Trabajo estimado: 15 horas.

Número: 2.2.2.1.

Nombre: Programación de herramientas básicas.

Descripción: Programar herramientas básicas para el escaneo de redes.

Trabajo estimado: 10 horas.

Número: 2.2.2.2.

Nombre: Integración de NMap.

Descripción: Integrar el núcleo de NMap en la aplicación para poder hacer uso de toda su funcionalidad.

Trabajo estimado: 10 horas.

Número: 2.2.2.3.

Nombre: Programación de parser de elementos.

Descripción: Elaborar un puente entre NMap y la aplicación para obtener los datos de NMap y poder usarlos en la aplicación de la manera más organizada posible.

Trabajo estimado: 20 horas.

Número: 2.2.2.4.

Nombre: Enlazado de datos con la GUI.

Descripción: Enlazar los datos con las diferentes vistas a través de diversos controladores, para poder visualizar e interactuar con ellos.

Trabajo estimado: 15 horas.

Número: 2.2.3.

Nombre: Testing.

Descripción: Una vez desarrollada la aplicación, realizar un amplio testeo para comprobar que funciona correctamente.

Trabajo estimado: 10 horas.

Número: 2.2.4.

Nombre: Corrección de bugs/errores.

Descripción: En base a los errores detectados en el testeo, implementar las correcciones a dichos fallos.

Trabajo estimado: 15 horas.

Número: 3.1.

Nombre: Integración del estado del arte.

Descripción: Integrar el estado del arte desarrollado dentro de la memoria.

Trabajo estimado: 5 horas.

Número: 3.2.

Nombre: Documentación de la aplicación desarrollada.

Descripción: Elaborar en base a todo el proceso de desarrollo una documentación clara sobre la aplicación e integrarla en la memoria.

Trabajo estimado: 15 horas.

Número: 3.3.

Nombre: Resumen de la memoria.

Descripción: Terminar la elaboración de la memoria, añadiendo las diferentes secciones necesarias y el formato correspondiente.

Trabajo estimado: 5 horas.

Número: 3.4.

Nombre: Elaborar presentación.

Descripción: Elaborar la presentación en diapositivas que se usará en la defensa ante el tribunal.

Trabajo estimado: 5 horas.

Número: 3.5.

Nombre: Preparación de la defensa.

Descripción: Preparar la defensa ante el tribunal en función a la documentación elaborada.

Trabajo estimado: 10 horas.

Número: 4.

Nombre: Reuniones periódicas.

Descripción: Reuniones periódicas con el director del TFG para llevar un control del desarrollo del proyecto.

Trabajo estimado: 15 horas.

2.2.4. Entregables

2.2.4.1. Fase 1

El entregable de la Fase 1 consistirá en un estado del arte redactado sobre el campo de la seguridad informática, enfocándose en el área del pentesting.

2.2.4.2. Fase 2

El entregable de la Fase 1 consistirá en una aplicación elaborada, que sirva como herramienta para el escaneo de redes informáticas. La aplicación, elaborada para Android, está correctamente empaquetada, con los posibles fallos corregidos y además dispondrá de una GUI sencilla de usar.

2.2.5. Cronograma

El cronograma completo, con las dos fases, como se muestra en las figuras 2.4, 2.5, 2.6 y 2.7 respectivamente.

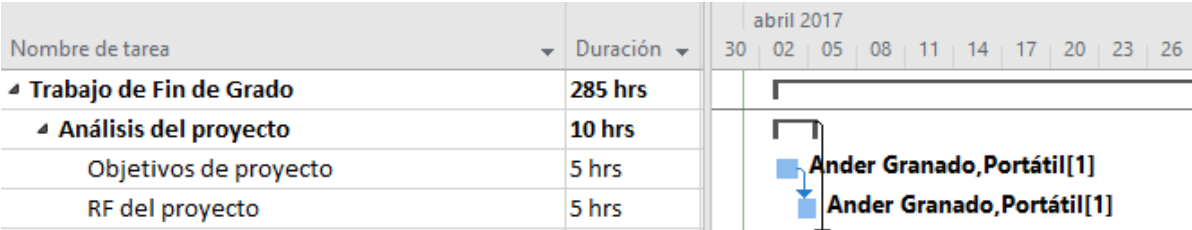


Figura 2.4.: Cronograma de la fase inicial

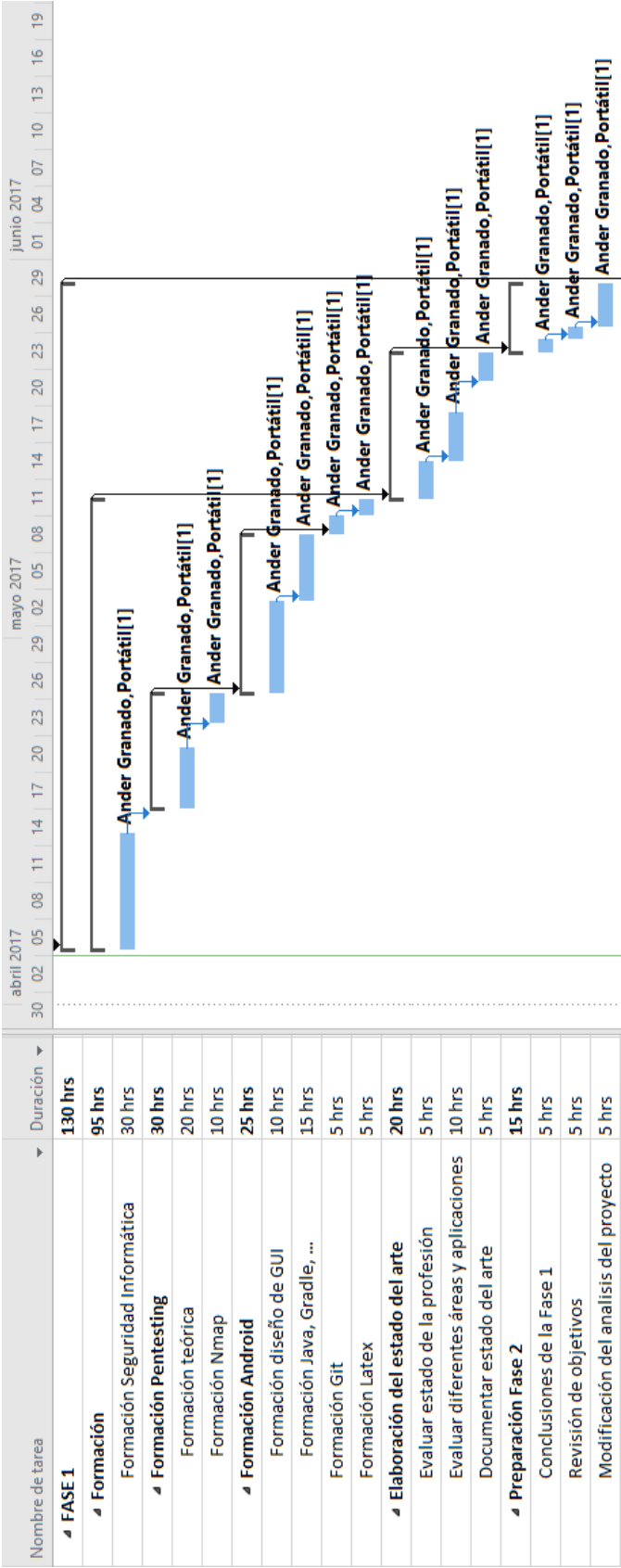


Figura 2.5.: Cronograma de la Fase 1

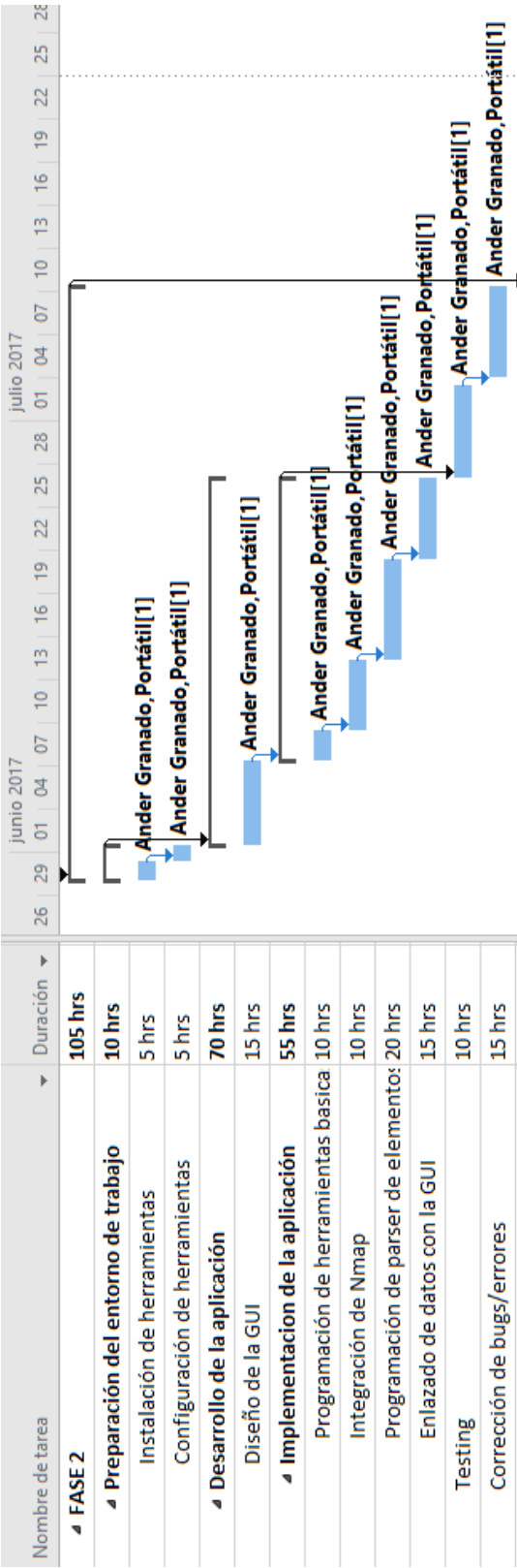


Figura 2.6.: Cronograma de la Fase 2

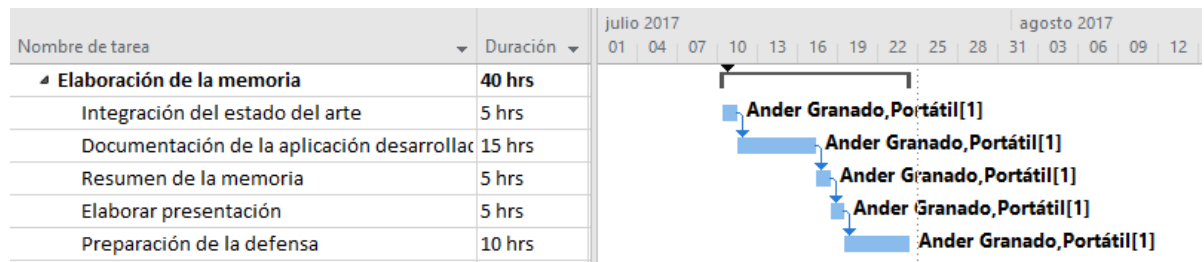


Figura 2.7.: Cronograma de la elaboración de la memoria

2.3. Gestión de costos

A la hora de elaborar un análisis de costos, con el objetivo de obtener un presupuesto, primero es necesario identificar todos los recursos que influyen en ese presupuesto. Por una parte, tendremos los recursos de trabajo, se decir empleados, y por otra parte los recursos materiales, tanto de software como de hardware. En la figura 2.8 se muestran tanto los recursos de trabajo como los recursos materiales de hardware.

		Nombre del	Tipo	Etiqueta de	Iniciales	Grupo	Capacidad	Tasa	Tasa horas	Costo/U:	Acumulado	Calendario base
1		Ander Granado	Trabajo		A		100%	14.00 €/hr	0.00 €/hr	0.00 €	Prorratio	Calendario Estudiantil
2		Portátil	Material		P			0.00 €		1.50 €	Prorratio	
3		BQ Aquaris M5	Material		BQ			0.00 €		1.50 €	Prorratio	

Figura 2.8.: Recursos de trabajo y materiales

Por otra parte, en la tabla 2.3 se muestra los recursos materiales de software utilizados.

Concepto	Coste
Windows 10	135,00 € [Microsoft(2017b)]
Project 2016	1.369,00 € [Microsoft(2017a)]
WBS Chart Pro	187,50 € [Critical Tools(2017)]
Ubuntu 16.04	0,00 €
TeX Studio	0,00 €
Android Studio	0,00 €
NMap	0,00 €
Git	0,00 €
GitHub	0,00 €

Cuadro 2.3.: Recursos materiales (software)

2.3.1. Presupuesto

El análisis de costes del proyecto se refleja en toda la información que aparecen entre la tabla 2.4 y la 2.10. Para ello se tienen en cuenta varios puntos. El primero, se tiene en cuenta un salario base de 14 €/h, correspondiente un salario estándar de un analista programador. Para los recursos materiales, se tiene en cuenta un costo por uso de 1,50 €. Estos costos por uso vienen dados por el gasto que generan debido al consumo energético, internet y otro tipo de gastos derivados de su utilización. Por otra parte se realiza una separación para los recursos de software, entre los que se incluye el propio Microsoft Project, software usado para la realización de todo el análisis de viabilidad y la gestión del proyecto. Para el cálculo de las amortizaciones se considera un tiempo de amortización de 3 años, el cual traducido en horas vendría a ser 4800 horas ($3 \text{ años} * 200 \text{ días laborables} * 8 \text{ horas} = 4800 \text{ horas}$).

Concepto	Coste
Ander Granado	14,00 €/h

Cuadro 2.4.: Recursos de trabajo

Concepto	Coste
Ordenador portátil	700,00 €
BQ Aquaris M5	250,00 €

Cuadro 2.5.: Recursos materiales (hardware)

Concepto	Coste	Numero de licencias
Windows 10	135,00 € [Microsoft(2017b)]	1
Project 2016	1369,00 € [Microsoft(2017a)]	1
WBS Chart Pro	187,50 € [Critical Tools(2017)]	1
Ubuntu 16.04	0,00 €	1
TeX Studio	0,00 €	1
Android Studio	0,00 €	1
NMap	0,00 €	1
Git	0,00 €	1
GitHub	0,00 €	1

Cuadro 2.6.: Recursos materiales (software)

Concepto	Trabajo (h)	Trabajo horas extra	Coste	Coste horas extra	Importe
Ander Granado	285	0	14,00 €/h	0	4.044,00 €

Cuadro 2.7.: Costo de recursos de trabajo

Concepto	Unidades	Coste	Importe
Ordenador portátil	1	1,50 €/uso	29 x 1,50 € = 43,50 €
BQ Aquaris M5	1	1,50 €/uso	7 x 1,50 € = 10,50 €
TOTAL			54,00 €

Cuadro 2.8.: Costo de recursos materiales

Concepto	Coste unitario	T. de amortización	C. U. amortización	T. de uso	Importe
Ordenador portátil	700,00 €	4800 horas	0,145833 €	285 h	41,57 €
BQ Aquaris M5	250,00 €	4800 horas	0,052083 €	25 h	1,30 €
Windows 10	135,00 €	4800 horas	0,028125 €	180 h	5,06 €
Microsoft Project 2016	1.369,00 €	4800 horas	0,285208 €	25 h	7,13 €
WBS Chart Pro	187,50 €	4800 horas	0,039062 €	5 h	0,19 €
Ubuntu 16.04	0,00 €	4800 horas	0,000000 €	105 h	0,00 €
TeX Studio	0,00 €	4800 horas	0,000000 €	40 h	0,00 €
Android Studio	0,00 €	4800 horas	0,000000 €	105 h	0,00 €
NMap	0,00 €	4800 horas	0,000000 €	30 h	0,00 €
Git	0,00 €	4800 horas	0,000000 €	145 h	0,00 €
GitHub	0,00 €	4800 horas	0,000000 €	145 h	0,00 €
TOTAL					55,25 €

Cuadro 2.9.: Amortizaciones de hardware y de software

Concepto	Importe
Recursos de Trabajo (R.T.)	4.044,00 €
Recursos Materiales (R.M.)	54,00 €
Costo fijo	0,00 €
Amorizaciones	55,25 €
SUMA	4.153,25 €
Gastos generales (10 %)	415,32 €
Beneficio (15 %)	622,99 €
SUBTOTAL	5.191,56 €
IVA (21 %)	1.090.23 €
TOTAL	6281.79 €

Cuadro 2.10.: Total presupuesto

Con esto se concluye que, tal como figura en la tabla 2.10, el coste del proyecto asciende a la cantidad de seis mil doscientos ochenta y cinco con setenta y nueve euros (6281.79 €).

2.4. Gestión de riesgos

En este apartado se identifican y analizan las diferentes amenazas que puedan llegar a impedir el correcto desarrollo del proyecto, haciendo que éste se retrase. Para ello primero se identificarán los diferentes riesgos que puedan existir y se indicará su peligrosidad. La peligrosidad es un valor cualitativo que indica cuánto puede afectar ese riesgo al proyecto.

Se han identificado los siguientes riesgos, los cuales se muestran en la tabla 2.11.

Riesgo	Peligrosidad
Pérdida de información	Alta
Enfermedades	Alta
Dificultades en la implementación de la aplicación	Alta
Dedicación no exclusiva al trabajo	Media
Averías o problemas técnicos con los recursos materiales	Media
Cambios o ampliación de requisitos	Media
Planificación muy optimista	Media

Cuadro 2.11.: Enumeración de riesgos del proyecto

2.4.1. Explicación y plan de contingencia

Tras haber identificado los riesgos inherentes al proyecto en la tabla 2.11, se hace mayor hincapié en los detalles de dicho riesgos, elaborando una descripción más detallada y analizando su probabilidad. La probabilidad de cada riesgo se muestra de manera cualitativa, ya que aportar un valor numérico concreto en este tipo de casos resulta bastante complicado. También, junto a lo mencionado, se añade para cada riesgo un plan de contingencia. El plan de contingencia consiste básicamente en aportar medidas para afrontar dicho riesgo, tanto preventivas (para antes de que ocurra) como correctoras (para en caso de ocurrir).

Riesgo: Pérdida de información.

Descripción: Podría darse el caso de que parte de la información, como la memoria del proyecto, el estado del arte o el código de la aplicación se perdieran.

Probabilidad: Baja.

Peligrosidad: Alta.

Medidas preventivas: Uso de herramientas de control de versiones como Git junto a uso de herramientas cloud como GitHub o Dropbox.

Medidas correctoras: Recuperación de la información mediante herramientas de análisis de unidades.

Riesgo: Enfermedades.

Descripción: Podría suceder que el único recurso de trabajo contrajera una enfermedad o tuviera un accidente.

Probabilidad: Baja.

Peligrosidad: Alta.

Medidas preventivas: Ninguna.

Medidas correctoras: Usar horas fuera del calendario para corregir el retraso en el proyecto.

Riesgo: Dificultades en la implementación de la aplicación.

Descripción: Podría suceder que, durante la implementación de la aplicación, se dieran dificultades a nivel de programación a la hora de cumplir con los requisitos funcionales.

Probabilidad: Media.

Peligrosidad: Alta.

Medidas preventivas: Una formación sólida en las herramientas y tecnologías que se van a usar para el desarrollo de la aplicación.

Medidas correctoras: Replantear las tareas posteriores y dedicar un esfuerzo extra para el aprendizaje y refuerzo de las herramientas usadas.

Riesgo: Dedicación no exclusiva al trabajo.

Descripción: Podría suceder que, debido a exámenes u otros asuntos personales, el desarrollador no pudiera aportar toda la dedicación que requiere el proyecto.

Probabilidad: Media.

Peligrosidad: Media.

Medidas preventivas: No previsible.

Medidas correctoras: Cambiar calendario de trabajo aumentando las horas para subsanar los retrasos producidos por dicho riesgo.

Riesgo: Averías o problemas técnicos con los recursos materiales.

Descripción: Podría darse el caso de que alguno de los recursos materiales de hardware, como el ordenador o el teléfono móvil, sufran algún tipo de avería.

Probabilidad: Baja.

Peligrosidad: Media.

Medidas preventivas: No previsible.

Medidas correctoras: Uso de otros dispositivos para continuar con el desarrollo del proyecto, adquiridos mediante compra o préstamo.

Riesgo: Cambios o ampliación de requisitos.

Descripción: Podría darse el caso en el que, tras la conclusión de la Fase 1, se cambiarán requisitos funcionales de la aplicación a desarrollar o se añadirán nuevos requisitos funcionales.

Probabilidad: Media - Alta.

Peligrosidad: Media.

Medidas preventivas: Revisión constante de los requisitos funcionales durante todas

las fases del proyecto para minimizar el impacto que pueda causar los cambios en ellos.

Medidas correctoras: Adaptar el calendario las tareas posteriores y el calendario de trabajo actual.

Riesgo: Planificación muy optimista.

Descripción: Puede darse el caso de que la planificación elaborada para el proyecto sea demasiado optimista y no haya tenido en cuenta ciertos aspectos más concretos del desarrollo del proyecto.

Probabilidad: Media.

Peligrosidad: Media.

Medidas preventivas: No previsible.

Medidas correctoras: Adaptar, y en caso de que fuera necesario retrasar, la fecha final del proyecto para dar cabida en la planificación toda esa duración extra no prevista.

II

Fase 1: Estado del Arte de la Seguridad Informática

Conceptos Generales

ESTADO DEL ARTE EN PROGRESO AQUI:

https://docs.google.com/document/d/1Zduryv6S6K_mwBTUrqA1NTov28VGW_3DHoxPQBBf608/edit?usp=sharing

Lo primero para hablar sobre seguridad informática es explicar que es (lógico no?). Diferenciar entre seguridad informática y seguridad de la información y definir y explicar ligeramente los términos mas usados en el campo.

3.1. Seguridad Informática

3.2. Seguridad de la Información

3.2.1. Diferencias respecto a la Seguridad Informática

3.3. Servicios de la Seguridad de la Información

3.3.1. Confidencialidad, Integridad y Disponibilidad

3.3.2. Otros servicios

3.3.2.1. Autenticación

3.3.2.2. Anonimato

3.3.2.3. Protección a la réplica

Aplicaciones de la Seguridad Informática

Después de hablar de los conceptos básicos sobre la seguridad informática, en este capítulo se profundiza en las diferentes aplicaciones que tiene y en que áreas. En definitiva, explicar para que se usa donde se usa, también un poco cómo se usa en cada área y porque es tan importante su uso.

Pentesting

En esta sección se profundiza sobre el Pentesting, que al final es el área de la seguridad informática sobre la que va la aplicación, el hecho de obtener información de sistemas y redes.

5.1. Objetivos

5.2. Partes

5.2.1. Recogida de información

5.2.1.1. Internal Footprinting

5.2.1.2. External Footprinting

5.2.2. Análisis de vulnerabilidades

5.2.3. Ataques de penetración

5.2.3.1. Ataques de contraseñas

5.2.3.2. Exploits

5.2.3.3. Auditoría de Aplicaciones Web

5.2.3.4. Ataques a redes

Ataques Wireless

Conclusiones

En este punto se extraen las conclusiones de todos los puntos anteriores, encarando el capítulo a elaborar la aplicación

III

Fase 2: Desarrollo de la aplicación

Introducción

Explicar tras el estado del arte en que va a consistir la aplicación.

Tecnologías y herramientas

Explicar las tecnologías que se van a usar para desarrollar la aplicación. He metido este punto aquí, en la FASE 2 ya que va mas ligado a ella, y ponerlo justo antes del desarrollo en sí me parece correcto.

8.1. Kali Linux

8.2. NMap

8.3. Android

8.3.1. Android SDK

8.3.2. Android Studio

8.4. Otras herramientas

8.4.1. Git

Desarrollo de la aplicación

Aquí ya al lío. Explicar como va a ser la GUI y como se han programado los diferentes aspectos de la app.

Testeo y corrección de errores

Tras programarlo todo pueden salir fallos, hay que hacer pruebas, e incluso habrá que cambiar cosas o añadir cosas nuevas. Todo eso va explicado aquí. NO CONFUNDIR con añadir nuevos requisitos funcionales o que se haya alargado el tiempo, eso va en la última parte, en la de conclusiones

IV

Análisis y conclusiones del Trabajo

V

Apéndice

Bibliografía

- [Critical Tools(2017)] Critical Tools, 2017. URL <https://store.criticaltools.com/>.
- [El Mundo(2011)] El Mundo, June 2011. URL <http://www.elmundo.es/elmundo/2011/06/09/navegante/1307619252.html>.
- [GitHub, Inc(2017)] GitHub, Inc, 2017. URL <https://github.com/showcases/security>.
- [Microsoft(2017a)] Microsoft, 2017a. URL https://www.microsoftstore.com/store/mseea/es_ES/pdp/Project-Profesional-2016/productID.324452600.
- [Microsoft(2017b)] Microsoft, 2017b. URL https://www.microsoftstore.com/store/mseea/es_ES/pdp/Windows-10-Home/productID.320437800.
- [Oficina de Seguridad del Internauta(2017)] Oficina de Seguridad del Internauta, 2017. URL <https://www.osi.es/es/herramientas-gratuitas>.