



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

**GRADO EN INGENIERÍA INFORMÁTICA
TECNOLOGÍA ESPECÍFICA DE INGENIERÍA DEL SOFTWARE**

TRABAJO FIN DE GRADO

**Herramienta para la evaluación de la seguridad de Bases de
Datos utilizando técnicas de Big Data.**

Julio Moreno García-Nieto

Septiembre, 2015



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

**DEPARTAMENTO DE TECNOLOGÍAS Y SISTEMAS DE
INFORMACIÓN**

TECNOLOGÍA ESPECÍFICA DE INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE GRADO

**Herramienta para la evaluación de la seguridad de Bases de
Datos utilizando técnicas de Big Data.**

Autor: Julio Moreno García-Nieto

Director: Manuel Ángel Serrano Martín

Septiembre, 2015

PÁGINA DE CALIFICACIÓN

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

En los días que vivimos, la información se ha convertido en unos de los activos más importantes para las empresas, por ello, los malhechores han convertido el robo, manipulación o borrado de información en uno de sus objetivos principales.

Para almacenar y poder consultar dicha información de forma rápida y sencilla surgió el concepto de bases de datos. Actualmente, una gran parte de la información del mundo se haya almacenada en bases de datos, por tanto, garantizar la seguridad de las mismas se convierte en una prioridad.

Para garantizar dicha seguridad, numerosos organismos e investigadores han creado una serie de estándares y técnicas que permiten la creación de sistemas seguros. Por ejemplo, la familia de normas ISO 27000 o MAGERIT.

El objetivo de este trabajo de fin de grado (TFG) será estudiar dichos estándares y mecanismos e implementar una serie de evaluaciones de seguridad que se realizarán mediante una herramienta web. Para ello, se hará uso de la tecnología de Big Data que permitirá estudiar la gran cantidad de datos que puede contener una base de datos relacional.

Abstract

Nowadays, information is one of the most important assets for the companies. Because of that, criminals decide that the theft, manipulation and erasing information will be one of their principal targets.

For storing and consulting information easily and quickly, databases appeared. Now, a big part of the information in the world is stored in databases, so preserve the security of them has been became a priority.

For guaranteeing this security, a lot of organizations and researchers have created numerous standards and techniques that allow the creation of secure systems. For example, the ISO 27000 family rules or MAGERIT.

The focus of this bachelor dissertation is the study of these standards and techniques for creating some evaluation that will be realized on a web application. For that purpose, I will use the Big Data technology that allow for studying the big amount of data that a relational database can contain.

X

A mis padres, por todo.

*A mis amigos de verdad, los que están ahí siempre,
ellos saben quienes son.*

Agradecimientos

Me gustaría empezar agradeciendo muy especialmente a mis padres, Julián e Isabel, por todo el apoyo mostrado en cualquier momento, además de ser un modelo en el que fijarme para ser mejor persona. También a mi familia, por mantenerse siempre a mi lado.

A Manuel Serrano por su confianza en mí para realizar este proyecto, siempre teniendo unas palabras de apoyo y un humor a prueba de bombas. A Ismael Caballero por haber sido en algunos momentos un “padre académico” al que acudir cuando había algún problema. A Eduardo Fernández-Medina por haberme proporcionado los medios necesarios para completar este proyecto.

A Dami, Clara y Patri por ser un punto de apoyo en los buenos y malos tiempos; si los amigos se eligen, yo he elegido bien.

A los “hachazines” de Paco y Javi, y a Elena por no ser sólo unos compañeros de clase, sino unos amigos en los que sé que puedo confiar y echar unas risas en cualquier momento.

A Cirebits y a la gente extraordinaria que la hace posible por permitirme obtener una experiencia inolvidable, no sólo a nivel profesional, sino también personal.

Al Grupo Alarcos, por sus consejos y su buen ambiente de trabajo, con esta gente da gusto ir a currar.

A los profesores que sienten pasión por su profesión y la contagian a sus alumnos, los que consiguen que vayas a clase con ganas de seguir aprendiendo.

A todos aquellos que me han acompañado en esta etapa de mi vida que se cierra con este TFG, si os pusiera a todos, me faltarían páginas.

Gracias, de verdad.

Julio.

Índice general

1	Introducción.....	1
1.1	Estructura del documento	5
2	Objetivos del TFG	7
2.1	Objetivo principal	7
3	Antecedentes, estado de la cuestión	9
3.1	Seguridad de la información	9
3.1.1	Familia de normas ISO/IEC 27000	11
3.1.2	COBIT 5	13
3.1.3	MAGERIT	15
3.2	Seguridad en bases de datos.....	17
3.2.1	Control de acceso.....	17
3.2.2	Cifrado de datos.....	18
3.3	Metodología para el diseño de bases de datos seguras	19
4	Método de Trabajo.....	21
4.1	Scrum	21
4.1.1	Roles	22
4.1.2	Sprint	23
4.1.3	Componentes de Scrum	24
4.2	Desarrollo basado en pruebas (TDD)	25
4.2.1	Ciclo de desarrollo de TDD.....	25
4.2.2	Ventajas de usar TDD.....	25
4.3	Kanban	26
4.4	Marco tecnológico para el desarrollo del proyecto.....	27
4.4.1	Herramientas de diseño	27
4.4.2	Herramientas de gestión del proyecto	28
4.4.3	Herramientas, tecnologías y frameworks para el desarrollo software.....	29
4.4.4	Herramientas para la gestión de bases de datos.....	33
4.4.5	Herramientas para realizar la documentación	35
4.4.6	Herramienta para el uso de máquinas virtuales	36
5	Resultados	39
5.1	Sprint 0.....	39
5.1.1	Gestión de recursos humanos	39
5.1.2	Alcance del proyecto	40
5.1.3	Plan de proyecto	41
5.1.4	Gestión temporal del proyecto.....	45
5.1.5	Gestión de las comunicaciones.....	47
5.1.6	Gestión de recursos.....	48

5.1.7	Gestión de riesgos.....	48
5.1.8	Gestión de costes.	50
5.2	Sprint 1.....	51
5.2.1	Refinamiento de la Pila del Producto	51
5.2.2	Planificación del Sprint	51
5.2.3	Desarrollo de la historia de usuario: <i>Añadir una base de datos para su evaluación</i>	52
5.2.4	Revisión del Sprint	55
5.3	Sprint 2.....	55
5.3.1	Refinamiento de la Pila del Producto	55
5.3.2	Planificación del Sprint	56
5.3.3	Desarrollo de la historia de usuario: <i>Disponer de diccionario en el sistema</i>	57
5.3.4	Desarrollo de la historia de usuario: <i>Comprobar datos de columna se encuentran cifrados</i>	58
5.3.5	Revisión del Sprint	61
5.4	Sprint 3.....	62
5.4.1	Refinamiento de la Pila del Producto	62
5.4.2	Planificación del Sprint	62
5.4.3	Desarrollo de historia de usuario: <i>Comprobar permisos de usuarios</i>	63
5.4.4	Revisión del Sprint	65
5.5	Sprint 4.....	66
5.5.1	Refinamiento de la Pila del Producto	66
5.5.2	Planificación del Sprint	66
5.5.3	Desarrollo de historia de usuario: <i>Comprobar número máximo de accesos de un usuario</i>	67
5.5.4	Revisión del Sprint	69
5.6	Sprint 5.....	70
5.6.1	Refinamiento de la Pila del Producto	70
5.6.2	Planificación del Sprint	70
5.6.3	Desarrollo de historia de usuario: <i>Introducir archivo con los usuarios a comprobar</i>	72
5.6.4	Desarrollo de historia de usuario: <i>Comprobar si a los usuarios eliminados se les han revocado los permisos</i>	72
5.6.5	Revisión del Sprint	74
5.7	Sprint 6.....	75
5.7.1	Refinamiento de la Pila del Producto	75
5.7.2	Planificación del Sprint	75
5.7.3	Desarrollo de historia de usuario: <i>Introducir archivo log</i>	77
5.7.4	Desarrollo de historia de usuario: <i>Introducir archivo con horarios de los usuarios</i>	78

5.7.5	Desarrollo de historia de historia de usuario: <i>Evaluar accesos de usuario en sus horas de trabajo</i>	78
5.7.6	Revisión del Sprint	81
5.8	Sprint 7.....	81
5.8.1	Refinamiento de la Pila del Producto	81
5.8.2	Planificación del Sprint	82
5.8.3	Desarrollo de historia de usuario: <i>Evaluar accesos fallidos a la base de datos</i>	82
5.8.4	Revisión del Sprint	85
5.9	Sprint 8.....	85
5.9.1	Refinamiento de la Pila del Producto	85
5.9.2	Planificación del Sprint	85
5.9.3	Desarrollo de historia de usuario: <i>Desarrollar interfaz web</i>	87
5.9.4	Desarrollo de historia de usuario: <i>Implementar aplicación web</i>	91
5.9.5	Revisión del Sprint	95
5.10	Sprint 9.....	96
5.10.1	Refinamiento de la Pila del Producto	96
5.10.2	Planificación del Sprint	96
5.10.3	Desarrollo de historia de usuario: <i>Representar los resultados con gráficos</i>	97
5.10.4	Desarrollo de historia de usuario: <i>Obtener pdf del informe final</i>	99
5.10.5	Revisión del Sprint	100
6	Conclusiones y propuestas.....	103
6.1	Análisis de consecución de los objetivos del proyecto	103
6.2	Propuestas de trabajos futuros	104
6.3	Otros datos de interés.....	104
6.4	Opinión personal.....	105
Bibliografía.....	107	
Anexo A. Manual de usuario	111	
Anexo B. Manual de instalación de la herramienta	121	
Anexo C. Acrónimos.....	123	

Índice de figuras

Figura 1.1 Dimensiones de seguridad	2
Figura 1.2 Estimación de información generada por año	3
Figura 1.3 Las 5 Vs de Big Data. Adaptada de [14].	4
Figura 2.1 Arquitectura general de la herramienta	8
Figura 3.1 Madurez en procesos de gestión de seguridad. Adaptado de [19]	10
Figura 3.2 Familia de COBIT 5 [9]	14
Figura 3.3 Principios de COBIT [9]	15
Figura 3.4 Ciclo PDCA [8]	16
Figura 3.5 Coste per cápita principales causas de pérdida de información [24].....	18
Figura 4.1 Componentes de Scrum.....	24
Figura 4.2 Esquema de funcionamiento de TDD.....	26
Figura 4.3 Tablero típico de Kanban	27
Figura 4.4 Herramientas de diseño	28
Figura 4.5 Herramientas de gestión del proyecto	29
Figura 4.6 Arquitectura de Hadoop	30
Figura 4.7 Herramientas para la gestión de bases de datos.....	35
Figura 4.8 Herramientas para realizar la documentación	36
Figura 4.9 Herramienta para el uso de máquinas virtuales	36
Figura 4.10 Resumen de las herramientas usadas en el proyecto	37
Figura 5.1 Diagrama de Gantt.....	46
Figura 5.2 Ejemplo de tablero en Trello (Sprint 6).....	47
Figura 5.3 Tablero general para gestionar el desarrollo de los sprints	48
Figura 5.4 HDFS funcionando con las carpetas "input" y "output".....	52
Figura 5.5 Ejemplo de uso de Sqoop	54
Figura 5.6 Resultados de ejecutar el comando anterior	54
Figura 5.7 Resumen Sprint 1	55
Figura 5.8 Funcionamiento de MapReduce	58
Figura 5.9 Funcionamiento evaluación 1: cifrado	61
Figura 5.10 SELECT sobre la tabla USER_PRIVILEGES	63
Figura 5.11 Funcionamiento de la evaluación 2: permisos.....	65
Figura 5.12 Máximo de conexiones por usuario.....	68
Figura 5.13 Archivo con los usuarios eliminados del sistema.....	72
Figura 5.14 Funcionamiento de la evaluación 4: usuarios eliminados	74
Figura 5.15 Ejemplo de archivo de log	77
Figura 5.16 Funcionamiento evaluación 5: accesos en horario permitido.....	80
Figura 5.17 Funcionamiento evaluación 6: accesos incorrectos.....	84
Figura 5.18 Prototipo pestaña de configuración	87
Figura 5.19 Prototipo pestaña de evaluaciones.....	88

Figura 5.20 Prototipo pestaña de informe final	88
Figura 5.21 Flujo de datos en una aplicación que usa stack MEAN [74].....	89
Figura 5.22 Interfaz final pestaña de configuración	90
Figura 5.23 Interfaz pestaña de evaluaciones	90
Figura 5.24 Interfaz pestaña de informe final	91
Figura 5.25 Base de datos de evaluaciones alojada en MongoLab	91
Figura 5.26 Interfaz final pestaña de evaluaciones	92
Figura 5.27 Ejemplos de pantallas de evaluaciones.....	93
Figura 5.28 Flujo de datos para la ejecución de un script.....	95
Figura 5.29 Informe final de la herramienta	99
Figura 5.30 Informe final en PDF.....	100
Figura A.1 Pestaña inicial de configuración.....	111
Figura A.2 Pestaña configuración completada	112
Figura A.3 Evaluaciones deshabilitadas sin archivo de log.....	112
Figura A.4 Listado de evaluaciones.....	113
Figura A.5 Evaluación 1	113
Figura A.6 Evaluación finalizada con éxito.....	114
Figura A.7 Evaluación 2	114
Figura A.8 Evaluación 3	115
Figura A.9 Evaluación 4	115
Figura A.10 Archivo para la evaluación 4	116
Figura A.11 Evaluación 5 en proceso	116
Figura A.12 Evaluación 6	117
Figura A.13 Archivo para la evaluación 6	117
Figura A.14 Evaluaciones realizadas.....	117
Figura A.15 Informe final	118
Figura A.16 Informe final en PDF.....	119

Índice de tablas

Tabla 2.1 Objetivos parciales del TFG	8
Tabla 3.1 Áreas de control de la ISO 27002.....	13
Tabla 5.1 Evaluaciones a realizar	40
Tabla 5.2 Pila de producto priorizada	42
Tabla 5.3 Plan de proyecto detallado	43
Tabla 5.4 Historias de usuario del sistema	45
Tabla 5.5 Análisis de riesgos	50
Tabla 5.6 Gestión de costes	50
Tabla 5.7 Historia de usuario "Añadir y procesar bases de datos"	51
Tabla 5.8 Historia de usuario "Tener un diccionario en el sistema".....	56
Tabla 5.9 Historia de usuario "Comprobar qué datos de una columna se encuentran cifrados"	57
Tabla 5.10 Historia de usuario "Comprobar permisos de usuarios"	63
Tabla 5.11 Historia de usuario "Comprobar máximo de conexiones de usuario".....	67
Tabla 5.12 Historia de usuario "Introducir archivo con usuarios eliminados"	71
Tabla 5.13 Historia de usuario "Comprobar qué usuarios eliminados no disponen de permisos"	71
Tabla 5.14 Historia de usuario "Introducir archivo de log"	76
Tabla 5.15 Historia de usuario "Introducir archivo con horario de los usuarios".....	76
Tabla 5.16 Historia de usuario "Evaluar que los usuarios sólo acceden en sus horas de trabajo"	77
Tabla 5.17 Historia de usuario "Evaluar accesos fallidos a la base de datos"	82
Tabla 5.18 Historia de usuario "Desarrollar interfaz web".....	86
Tabla 5.19 Historia de usuario "Implementar aplicación web"	87
Tabla 5.20 Historia de usuario "Representar los resultados con gráficos"	97
Tabla 5.21 Historia de usuario "Obtener pdf del informe final"	97
Tabla 5.22 Gráficos a utilizar en función de la evaluación.....	98
Tabla 6.1 Consecución de objetivos parciales	103

Índice de listados

Listado 5.1 Pruebas unitarias función map (evaluación 1)	59
Listado 5.2 Función map de la evaluación 1	59
Listado 5.3 Pruebas unitarias de la función reduce (evaluación 1)	60
Listado 5.4 Función reduce de la evaluación 1	60
Listado 5.5 Pruebas unitarias de la función map (evaluación 2)	64
Listado 5.6 Función map de la evaluación 2	64
Listado 5.7 Pruebas unitarias de la función reduce (evaluación 2)	64
Listado 5.8 Función reduce de la evaluación 2	65
Listado 5.9 Pruebas unitarias de la evaluación 3	68
Listado 5.10 Código correspondiente a la evaluación 3	69
Listado 5.11 Pruebas unitarias de la función map (evaluación 4)	72
Listado 5.12 Función map de la evaluación 4	73
Listado 5.13 Pruebas unitarias de la función reduce (evaluación 4).....	73
Listado 5.14 Función reduce de la evaluación 4.....	73
Listado 5.15 Pruebas unitarias de la función map (evaluación 5)	78
Listado 5.16 Función map de la evaluación 5	79
Listado 5.17 Pruebas unitarias de la función reduce (evaluación 5).....	79
Listado 5.18 Función reduce de la evaluación 5.....	80
Listado 5.19 Pruebas unitarias de la función map (evaluación 6)	83
Listado 5.20 Función map de la evaluación 6	83
Listado 5.21 Pruebas unitarias de la función reduce (evaluación 6).....	83
Listado 5.22 Función reduce de la evaluación 6.....	84
Listado 5.23 Ejemplo de los scripts de la evaluación 4	94
Listado 5.24 Método para crear el PDF	100

1 Introducción

En los años 60, la necesidad de almacenar grandes cantidades de información de una forma ordenada y estructurada, se estaba convirtiendo en una necesidad prioritaria. Además se quería que se pudiera acceder y buscar en dicha información almacenada, de una forma más rápida y efectiva. Como respuesta a esta necesidad, surgieron las bases de datos [1].

Una base de datos es una colección ordenada de datos, que normalmente están organizados para modelar aspectos de la realidad de manera que soporta procesos para acceder y modificar dicha información de forma cómoda y efectiva [2].

En la actualidad, las bases de datos se han convertido en un estándar en cualquier sistema de información, ya que, la información se ha convertido en un recurso esencial para desarrollar la actividad normal de cualquier persona, entidad o empresa.

Esto lo convierte en un activo tan importante como podrían ser el trabajo o los recursos económicos, e implica que debe ser protegido de forma adecuada. Para ello, surge la seguridad de la información [3].

La seguridad de la información incluye la gestión y aplicación de las medidas de seguridad apropiadas que toman en consideración un amplio rango de amenazas, con el objetivo de asegurar la continuidad y sostenibilidad del negocio, minimizando el impacto provocado en la información debido a incidentes de seguridad.

Para ello, se implementan una serie de controles, seleccionados a partir de un proceso de gestión de riesgos, esta serie de controles se pueden aplicar tanto a políticas, procesos, procedimientos, estructuras de negocio, software y hardware. Estos controles deben ser especificados, implementados, monitorizados, revisados y mejorados cuando sea necesario, para garantizar que se sigan cumpliendo [4].

Aun con estos controles, hablar de seguridad informática en términos absolutos es imposible y por ello, se habla más de fiabilidad del sistema que, en realidad es una relajación del primer término.

Existe una frase muy conocida en el mundo de la seguridad que describe esta problemática, la dijo Eugene Spafford (experto en seguridad de datos y profesor de ciencias informáticas en la Universidad Purdue): “El único sistema seguro es aquel que está apagado y desconectado, enterrado en un refugio de cemento, rodeado por gas venenoso y custodiado por guardianes bien pagados y muy bien armados. Aun así yo no apostaría por él” [5].

Por estos motivos, se considerara que un sistema es seguro si se pueden garantizar los principales tres dominios de la seguridad de la información [6]:

- **Confidencialidad:** el acceso a la información está limitado a usuarios autorizados. Cuando los datos son de carácter personal se le denomina privacidad.
- **Integridad:** los activos del sistema sólo pueden ser borrados o modificados para usuarios autorizados.
- **Disponibilidad:** el acceso a los activos en un tiempo razonable está garantizado para usuarios autorizados.



Figura 1.1 Dimensiones de seguridad

En el caso concreto que nos ocupa, el hecho de que una gran parte de la información del mundo se encuentre almacenada en distintos sistemas gestores de bases de datos, hace que sean muy comunes los ataques a dichas bases de datos para intentar conseguir información. Por ello, cada vez se convierte en una necesidad más prioritaria el controlar y evaluar la seguridad en nuestras bases de datos [7].

Esta problemática ha sido tratada por una gran cantidad de investigadores y organizaciones, que han dado lugar a distintos estándares y recomendaciones relacionadas con la seguridad de las bases de datos, algunos ejemplos pueden ser:

- MAGERIT (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) [8].
- COBIT (Control Objectives for Information and related Technology) [9].
- Las distintas normas de la familia ISO 27000 [4].

Todas estas publicaciones serán explicadas con mayor detalle en el capítulo dedicado al estado de la cuestión (capítulo 3).

Aun teniendo distintos estándares y metodologías para crear una base de datos segura, existe un problema a la hora de evaluar dicha seguridad de las bases de datos, y ese punto débil es precisamente lo que a su vez es su punto fuerte: la gran cantidad de datos que pueden almacenar [10].

Esto sucede debido a que en el mundo actual, la manera en la que los usuarios se relacionan con la tecnología ha evolucionado de una forma radical: web 2.0, redes sociales, multimedia, etc., junto a la presencia on-line perpetua gracias a los teléfonos inteligentes. Estos motivos hacen que cada vez se genere más y más información, estas cantidades de información eran impensables hace unos años.

Por ejemplo, se estima que hasta 2003 el ser humano había generado una cantidad de información equivalente a 5 Exabytes (Un exabyte son 10^{18} bytes) de contenido. Actualmente, se genera esa cantidad de información en cuestión de días [11].

En la figura 1.2, se puede observar cuánta información se generó por año y una estimación de los próximos años (En Zettabytes, 1 zettabyte son 10^{21} bytes), así se puede observar que la cantidad de datos está creciendo a un ritmo del 40% anual, con lo que se espera que se alcancen los 45 ZB en el año 2020.

Data is growing at a 40 percent compound annual rate, reaching nearly 45ZB by 2020

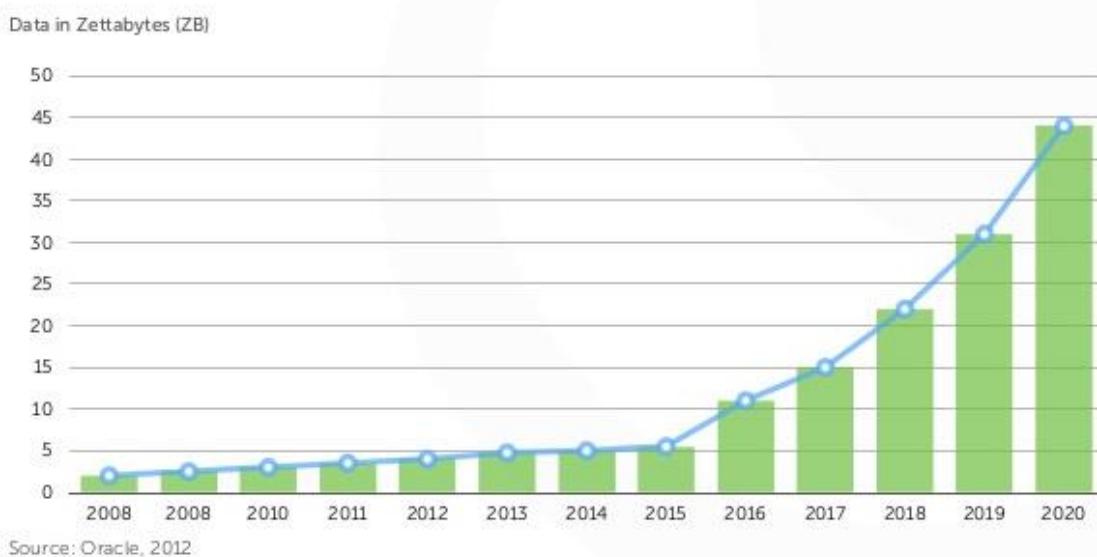


Figura 1.2 Estimación de información generada por año

Esta gran cantidad de datos hace que sea necesario un sistema capaz de procesarlos de forma correcta y rápida, una posible solución a esta necesidad de velocidad en un entorno heterogéneo y voluminoso es la utilización de técnicas de análisis de Big Data [12].

El término Big Data surgió para definir un nuevo paradigma para aplicaciones de datos, ha sido definido de muchas formas distintas pero el núcleo del paradigma Big Data es que permite manejar cantidades de datos muy grandes (volumen), que llegan muy rápido (velocidad) y cambian muy rápido (variable), contienen mucho ruido (veracidad) y además son muy diversas (variedad) como para ser procesadas con las técnicas tradicionales [13].

A estas características se las conoce como las 5 Vs del Big Data, en la figura 1.3 se puede ver un resumen de dichas propiedades.

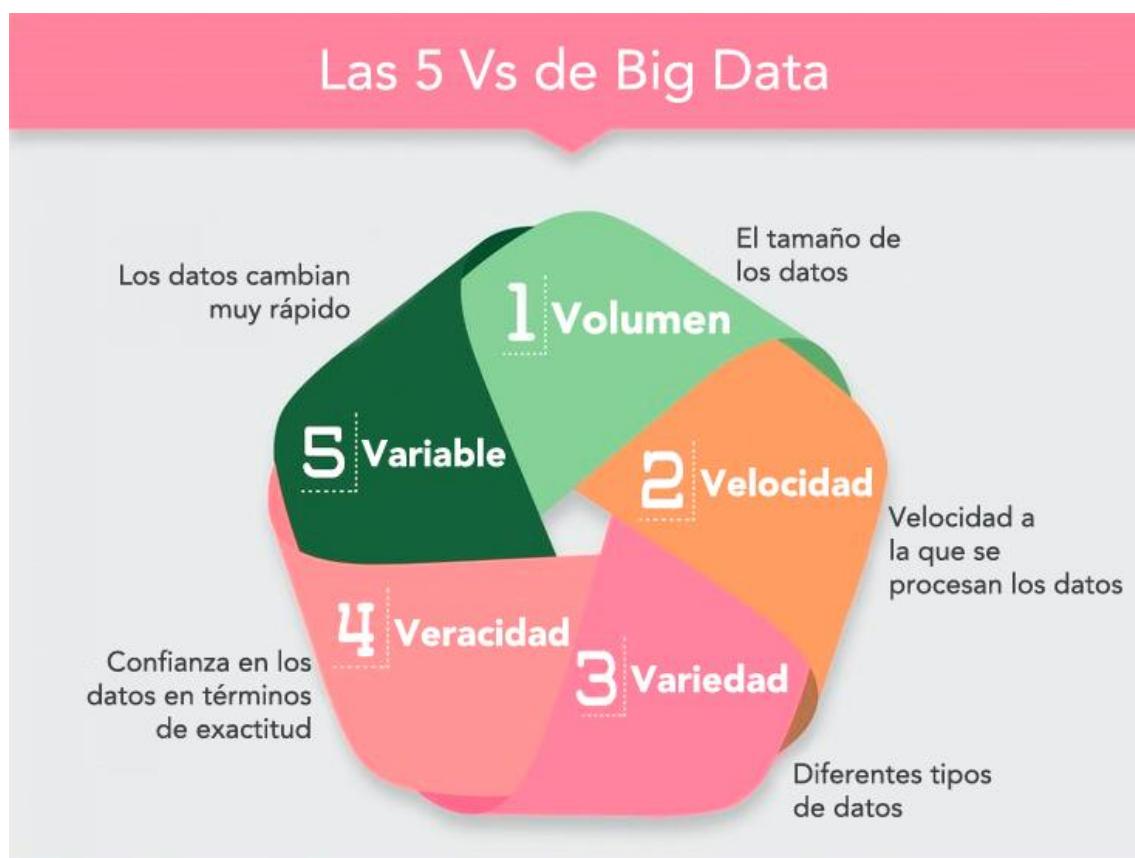


Figura 1.3 Las 5 Vs de Big Data. Adaptada de [14].

Esta capacidad de poder analizar grandes cantidades de datos de forma rápida y eficaz, permite encontrar relaciones entre los datos que, a primera vista, parecen no tener nada en común obteniendo una información de gran utilidad para las empresas u organizaciones que hacen uso de *Big Data*; es una herramienta perfecta para detectar sutiles correlaciones entre los datos, consiguiendo diagnósticos y pronósticos fiables en las distintas áreas de la empresa u organización [15].

Por tanto, estas características encajan con la dificultad encontrada a la hora de evaluar la seguridad de una base de datos y, por tanto, hacen posible realizar el objetivo principal de este trabajo fin de grado, el cual, será crear una herramienta web que nos permita realizar la evaluación de la seguridad en una base de datos usando para ello tecnologías basadas en Big Data.

1.1 Estructura del documento

La memoria de este trabajo de fin de grado se estructurará a través de capítulos cuyo contenido será el siguiente:

- Capítulo 1 – Introducción: breve introducción al tema a tratar durante el trabajo de fin de grado, además de explicación superficial de la herramienta a desarrollar.
- Capítulo 2 – Objetivos del TFG: vistazo al objetivo principal y a los objetivos parciales que se quieren cumplir con la realización de este TFG.
- Capítulo 3 – Antecedentes, estado de la cuestión: información relevante referida al tema concreto de la seguridad en bases de datos.
- Capítulo 4 – Método de trabajo: explicación de la metodología de trabajo utilizada para, tanto la gestión como la implementación del proyecto a desarrollar. También se indican las herramientas utilizadas para la realización del TFG.
- Capítulo 5 – Resultados: se exponen los resultados obtenidos tras la aplicación de las metodologías escogidas. También se indica qué pasos se han seguido para conseguir dichos resultados.
- Capítulo 6 - Conclusiones y propuestas: se explican las metas y competencias conseguidas tras la realización del TFG comparándolas con los objetivos expresados en el Capítulo 2 de la memoria. También se indican propuestas futuras para mejorar la solución propuesta.
- Bibliografía: Recopilación de toda la bibliografía utilizada para la elaboración de este TFG.

- Anexo A: Manual de usuario.
- Anexo B: Manual de instalación de la herramienta.
- Anexo C: Acrónimos.

2 Objetivos del TFG

En este capítulo se expone el objetivo principal de este TFG, así como los objetivos parciales que se intentarán conseguir con la realización de este trabajo.

2.1 Objetivo principal

El objetivo principal del TFG será el desarrollo de una aplicación web que permita evaluar la seguridad de una base de datos relacional.

Como se explicó en el capítulo 1, la seguridad se trata de una disciplina compuesta de diversos criterios como son la confidencialidad, la integridad, la disponibilidad o la autenticidad [16]; esto hace que sea una disciplina ambigua, ya que es difícil afirmar con rotundidad que un sistema es seguro, por ello el objetivo de este TFG es evaluar conceptos relacionados con seguridad en una base de datos, pudiendo ser extensible el número de criterios a evaluar por la herramienta en el futuro.

El sistema recibirá como entrada la base de datos a evaluar, siendo proporcionada por el cliente mediante un enlace, incluyendo los datos de acceso necesarios. Para poder evaluar la seguridad de dicha base de datos, el cliente especificará las reglas de negocio a evaluar. También podrá ser necesario proporcionar un *archivo de log* para la realización de algunas evaluaciones.

Para la evaluación de la seguridad de la base de datos, se implementarán una serie de controles parametrizables por el usuario de la herramienta. Estos controles se implementarán haciendo uso de un sistema basado en Big Data.

Finalmente como salida del sistema se obtendrá un informe detallado con las reglas de seguridad que cumple e incumple la base de datos introducida.

El objetivo principal del TFG será la realización de una herramienta web que permite la evaluación de la seguridad de una base de datos relacional mediante el uso de técnicas de Big Data.

En la figura 2.1 se puede ver un diagrama explicativo de la arquitectura del sistema que se pretende implantar en el presente Trabajo Fin de Grado.

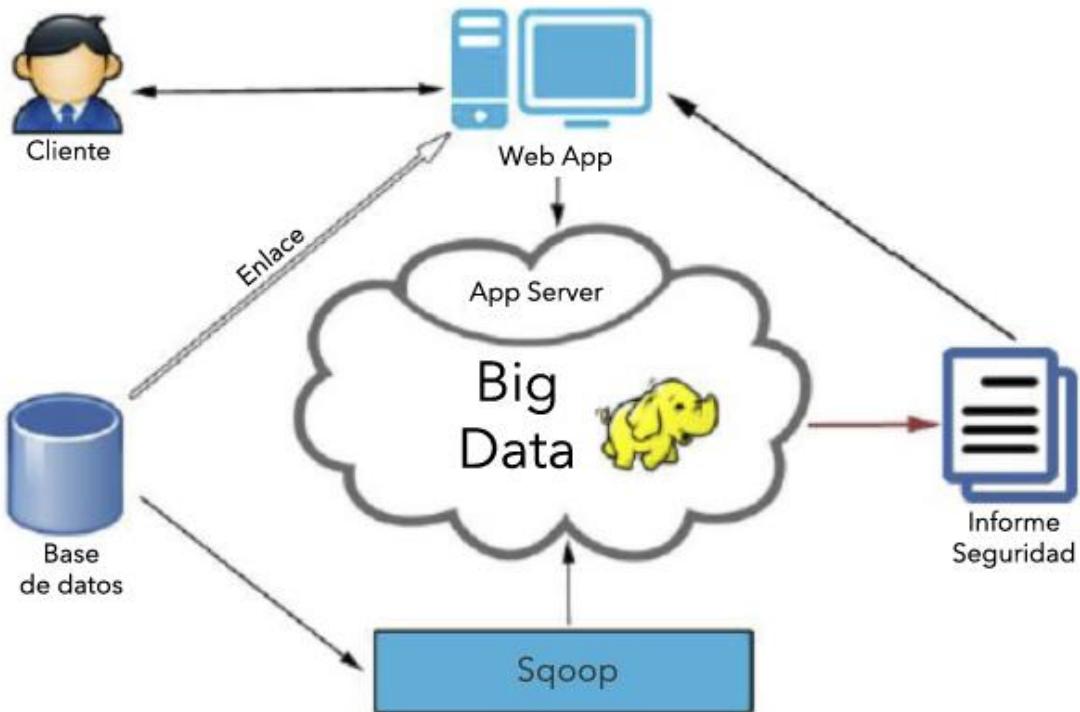


Figura 2.1 Arquitectura general de la herramienta

2.1.1. Objetivos parciales

En la tabla 2.1 se encuentra un resumen de los distintos objetivos parciales de los que consta el TFG y cuya consecución por separado, llevará al cumplimiento del objetivo principal del proyecto.

Id Objetivo	Objetivo
O.1	Estudiar los problemas típicos de seguridad de las bases de datos relacionales. Una vez conocidos, elegir las evaluaciones a realizar por la herramienta.
O.2	Introducir la información de una base de datos relacional en un sistema Big Data.
O.3	Implementación de las evaluaciones seleccionadas mediante técnicas de Big Data (en las que sean necesario).
O.4	Realización de una interfaz web que permita realizar todas las evaluaciones ya implementadas.
O.5	Generación de un informe final con los resultados obtenidos de ejecutar las evaluaciones.

Tabla 2.1 Objetivos parciales del TFG

3 Antecedentes, estado de la cuestión

En este capítulo se expone la información relevante que servirá como base para el desarrollo del TFG. Así, durante esta sección se expondrá la problemática de la seguridad en las TI, así como de forma más concreta a la seguridad relacionada con las bases de datos. Para ello, se tendrán en cuenta metodologías, modelos, tesis y artículos científicos.

3.1 Seguridad de la información

Tal y como se comentó durante el capítulo dedicado a la introducción, en la época en la que nos encontramos, en la cual, la información se ha convertido en uno de los recursos más importantes de las empresas, es indispensable que conseguir que estos datos no sean accesibles o modificables por gente ajena a la compañía sea una prioridad y un motivo de preocupación [3].

Según [17], la seguridad es “la calidad o estado de estar libre de peligro”, es decir, protección contra los adversarios (aquellos que nos pueden hacer daño tanto de forma intencionada como involuntaria). Para conseguir este nivel apropiado de seguridad, se establecen varios tipos de seguridad:

- **Seguridad física:** para proteger objetos o áreas físicas de usos no autorizados o mal uso.
- **Seguridad de personal:** para proteger al sistema de individuos o grupos de individuos que están autorizados a acceder a la organización y sus operaciones.
- **Seguridad de operaciones:** proteger los detalles de una operación en particular o de una serie de actividades.
- **Seguridad de comunicaciones:** para proteger los datos y las tecnologías propias de las comunicaciones.
- **Seguridad de red:** para proteger componentes, conexiones y contenidos de la red.
- **Seguridad de la información:** su objetivo es proteger la confidencialidad, integridad y disponibilidad de los recursos de información, tanto en su almacenamiento, su proceso o su transmisión. Esto se consigue mediante la aplicación de políticas, cursos de educación, entrenamiento y concienciación, y mediante tecnología.

La familia de estándares ISO/IEC 27000 [4] define esta seguridad de la información como el conjunto de políticas, procedimientos, guías, actividades y procesos, que se encuentran gestionadas por una organización y cuyo objetivo es el de proteger los activos de información.

Estos activos se encuentran expuestos a una gran cantidad de amenazas, así pues el objetivo de implementar seguridad en un sistema será reducir el impacto que pudiera tener la acción de dichas amenazas en el sistema. Un sistema seguro debe cumplir las tres dimensiones principales: confidencialidad, disponibilidad e integridad.

Estas amenazas pueden tener distintos orígenes [18]:

- Naturales: inundaciones, incendios, tormentas, etc.
- Agentes externos: virus informáticos, ataques de organizaciones criminales, robos, estafas, etc.
- Agentes internos: empleados descuidados con una formación inadecuada o descontentos, errores en la utilización de herramientas y recursos del sistema, etc.

Aun siendo conscientes de la importancia que tiene la seguridad para proteger sus activos, sucesivas encuestas demuestran que la implicación de las empresas en temas de seguridad no es todo lo buena que debería ser. Por ejemplo, en 2013 se realizó una encuesta sobre empresas de todos los sectores para comprobar el grado de madurez de los procesos de gestión de seguridad, obteniendo como resultado el gráfico mostrado en la figura 3.1.

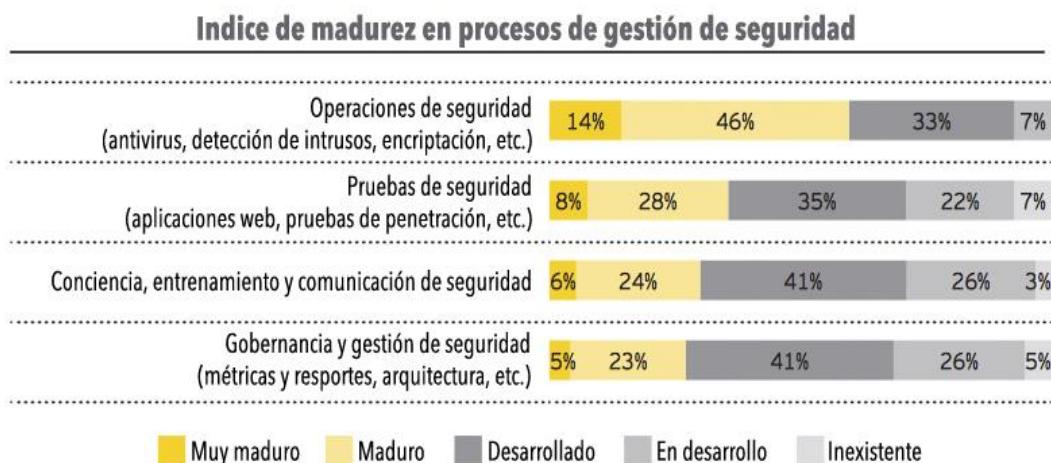


Figura 3.1 Madurez en procesos de gestión de seguridad. Adaptado de [19]

Como se puede ver el grado de madurez mostrado para las diferentes áreas encuestadas

no alcanza el nivel de “muy maduro” en la mayoría de los casos. Por ello, desde hace tiempo existen una serie de estándares y metodologías destinadas a mejorar la seguridad de los distintos sistemas.

3.1.1 Familia de normas ISO/IEC 27000

La familia de normas ISO/IEC 27000 [4] se trata de un conjunto de estándares desarrollados o en fase de desarrollo creados por ISO (International Organization for Standardization) e IEC (International Electrotechnical Commision), cuyo objetivo es proporcionar un marco para la gestión de la seguridad de la información utilizable por cualquier tipo de organización. Esta familia se encuentra formada principalmente por las siguientes normas [20]:

- **ISO 27000:** proporciona una visión general del tema de la seguridad de la información, crea un vocabulario común que servirá como base para el resto de normas. También se da una pequeña introducción a cada una de las normas que la componen.
- **ISO 27001:** proporciona los requisitos del sistema de gestión de seguridad de la información (SGSI).
- **ISO 27002:** es una guía de buenas prácticas que describe los objetivos de control y controles recomendables en cuanto a seguridad de la información. Contiene 39 objetivos de control y 133 controles, agrupados en 11 dominios.
- **ISO 27003:** es una guía que se centra en los aspectos críticos necesarios para el diseño e implementación con éxito de un SGSI siguiendo la norma ISO 27001. Describe todo el proceso desde la concepción hasta la puesta en marcha de planes de implementación.
- **ISO 27004:** guía para el desarrollo y utilización de métricas y técnicas de medida aplicables para determinar la eficacia de un SGSI y de los controles o grupos de controles implementados según ISO 27001.
- **ISO 27005:** proporciona directrices para la gestión de riesgo en la seguridad de la información. Apoya los conceptos generales especificados en la norma ISO 27001 y está diseñada para ayudar a la aplicación satisfactoria de la seguridad de la información.
- **ISO 27006:** especifica los requisitos para la acreditación de entidades de auditoría y certificación de sistemas de gestión de la seguridad de la información.
- **Otras normas de la familia:** aparte de las normas ya mencionadas, existen otra gran cantidad de normas que conforman esta familia como, por ejemplo, la ISO 27010 (guía para compartir información de forma segura entre organizaciones o sectores), la ISO 27018 (código de buenas prácticas en controles de protección de datos para cloud computing), la ISO 27032 (proporciona orientación para la

mejora del estado de la seguridad cibernética), la ISO 27033 (dedicada a la seguridad en redes), etc.

Una vez conocidas las normas ISO más importantes de la familia de ISO 27000, se va a definir más en detalle la referida a la guía de buenas prácticas y controles recomendables, es decir, la norma ISO 27002.

3.1.1.1 Norma ISO 27002

La normativa ISO 27002 [21] ofrece soluciones genéricas en el ámbito de la seguridad que pueden ser aplicables por cualquier empresa u organización. Estas recomendaciones se encuentran divididas en 11 áreas de control, resumidas en la siguiente tabla:

Área de control	Objetivos
Políticas de seguridad	<ul style="list-style-type: none">• Proveer gestión en la dirección y soporte de la seguridad de la información, de forma que se encuentre alineado con los requisitos de negocio, y con las leyes y regulaciones. Obteniendo un documento de políticas de seguridad.
Organización de la seguridad de la información	<ul style="list-style-type: none">• Establecer un entorno de gestión para iniciar y controlar la implementación y operación de la seguridad de la información a través de la organización.• Asegurar la seguridad del trabajo a distancia y el uso de dispositivos móviles.
Seguridad de recursos humanos	<ul style="list-style-type: none">• Asegurar que los empleados entienden sus responsabilidades y son adecuados para su puesto.• Asegurar que los empleados son conscientes y cumplen sus responsabilidades de seguridad de la información.• Proteger los intereses y recursos de la organización cuando un empleado ya no continua en su puesto.
Gestión de activos	<ul style="list-style-type: none">• Identificar los activos de la compañía y definir apropiadas responsabilidades de protección.• Asegurar que la información recibe un apropiado nivel de protección, de acuerdo con su importancia.• Prevenir la revelación, modificación o eliminación de información almacenada.
Control de accesos	<ul style="list-style-type: none">• Limitar el acceso a la información y a las instalaciones.• Asegurar accesos autorizados de usuarios y prevenir accesos no autorizados.• Hacer a los usuarios responsables de salvaguardar su información de autenticación.

Seguridad física y ambiental	<ul style="list-style-type: none"> Prevenir accesos físicos no autorizados, daño e interferencia con la información o los servicios de la organización. Prevenir perdida, daño o robo de activos.
Seguridad de operaciones y de comunicaciones	<ul style="list-style-type: none"> Asegurar que las operaciones de proceso de información se realicen de forma correcta y segura. Asegurar que la información se encuentra protegida contra malware. Disponer de backups para evitar la pérdida de datos. Monitorizar y guardar un archivo de log con las incidencias en el sistema. Asegurar la integridad de sistemas operacionales. Prevenir la explotación de las vulnerabilidades del sistema. Minimizar el impacto de las amenazas. Asegurar la protección de información en redes. Mantener la seguridad de la información transferida a través de la organización y con cualquier entidad externa.
Adquisición, desarrollo y mantenimiento de sistemas	<ul style="list-style-type: none"> Asegurar que la seguridad de la información se encuentra integrada en todo el ciclo de vida del sistema. Asegurar la seguridad de la información usada en pruebas. Asegurar el uso apropiado y efectivo de la criptografía para proteger la confidencialidad, autenticidad y/o integridad de la información.
Gestión de incidentes de seguridad de la información	<ul style="list-style-type: none"> Asegurar un enfoque efectivo y consistente para la gestión de incidentes relacionados con la seguridad de la información.
Gestión de la continuidad del negocio	<ul style="list-style-type: none"> La seguridad de la información debe ir incluida en los planes de gestión de continuidad del negocio.
Cumplimiento	<ul style="list-style-type: none"> Evitar agujeros legales, de estatutos u obligaciones contractuales relacionados con la seguridad de la información o con cualquier requisito de seguridad.

Tabla 3.1 Áreas de control de la ISO 27002

3.1.2 COBIT 5

COBIT 5 es un [9] marco de gobierno de las tecnologías de la información que proporciona una serie de herramientas para que la dirección pueda conectar los requerimientos de control con los aspectos técnicos y los riesgos del negocio. Permite el desarrollo de las políticas y buenas prácticas para el control de las tecnologías en toda la organización. Además hace hincapié en ayudar a las organizaciones a aumentar su valor gracias a las tecnologías y permite su alineación con los objetivos de negocio. COBIT 5

forma parte de una familia de productos que se puede ver en la siguiente figura.

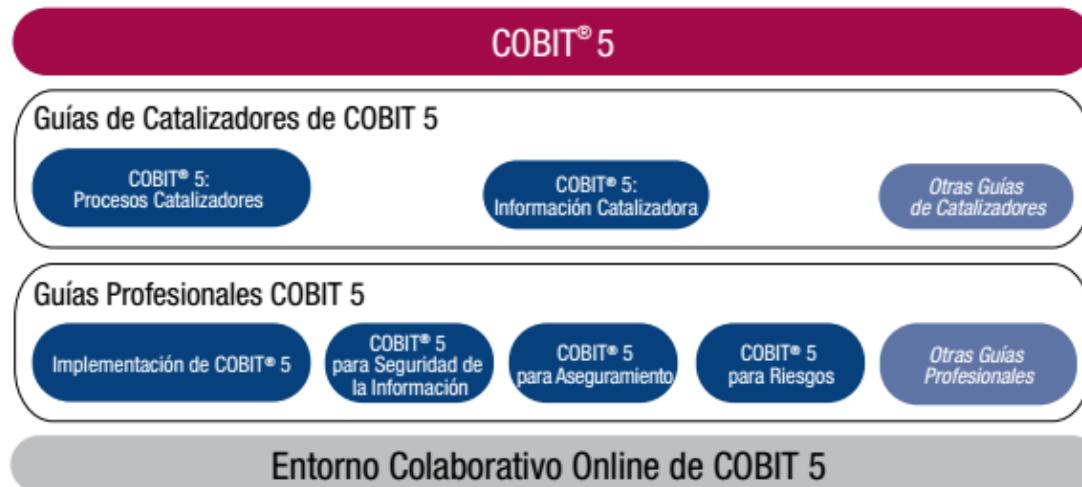


Figura 3.2 Familia de COBIT 5 [9]

Para integrar la seguridad en su modelo, COBIT 5 toma como base el modelo relacional que utiliza BMIS (Business Model for Information Security), incorporando su visión integral y sus componentes a la nueva versión presentando un enfoque orientado al negocio para la gestión de la seguridad de la información.

COBIT establece un lenguaje común para referirse a la protección de la información, cambiando la visión convencional de la inversión en seguridad de la información. Se encuentra basado en cinco principios claves para el gobierno y la gestión de las TI empresariales:

- **Principio 1. Satisfacer las necesidades de las partes interesadas:** las empresas existen para crear valor para sus partes interesadas manteniendo el equilibrio entre la consecución de beneficios y la optimización de los riesgos y uso de recursos. COBIT provee los procesos necesarios para permitir la creación de valor del negocio mediante el uso de las TI.
- **Principio 2. Cubrir la empresa extremo a extremo:** integra el gobierno y la gestión de TI en el gobierno corporativo, cubriendo todas las funciones y procesos dentro de la empresa.
- **Principio 3. Aplicar un marco de referencia único integrado:** hay muchos estándares y buenas prácticas relativos a las TI; COBIT se alinea a alto nivel con otros estándares y marcos de trabajo relevantes.
- **Principio 4. Hacer posible un enfoque holístico:** un gobierno y gestión de las TI de la empresa efectivo y eficiente requiere un enfoque holístico que tenga en cuenta varios componentes interactivos. COBIT define un conjunto de

catalizadores para apoyar la implementación de un sistema de gobierno y gestión global para las TI de la empresa.

- **Principio 5. Separar el gobierno de la gestión:** se establece una clara distinción entre gobierno y gestión. Estas dos disciplinas engloban diferentes tipos de actividades, requieren diferentes estructuras organizativas y tienen diferentes objetivos.



Figura 3.3 Principios de COBIT [9]

3.1.3 MAGERIT

MAGERIT [8] implementa el proceso de Gestión de Riesgos dentro de un marco de trabajo para que los órganos de gobierno tomen decisiones teniendo en cuenta los riesgos derivados del uso de tecnologías de la información. Para ello, dispone de dos tipos de elementos: un conjunto de guías y un panel de herramientas de apoyo.

Como se ha discutido en este capítulo, existen varias aproximaciones al problema de analizar los riesgos soportados por un sistema basado en las TI. Su objetivo es realizar el análisis de riesgos para saber cuán seguros o inseguros son los sistemas evaluados. El gran problema al que se enfrentan, es la gran cantidad de elementos a considerar, lo que implica que si no se hace bien, las conclusiones derivadas de este estudio serán poco de

fiar. Por ello, MAGERIT defiende una aproximación metódica que no deje lugar a la improvisación.

Estas tareas de análisis y tratamiento de los riesgos no son un fin en sí mismas sino que se encajan dentro de la actividad continua de la gestión de la seguridad. En coordinación con los objetivos, estrategia y política de la organización, las actividades relacionadas con el tratamiento de los riesgos permiten elaborar un plan de seguridad que, implementado y operado, satisfaga los objetivos propuestos con el nivel de riesgo aceptado.

La implantación de las medidas de seguridad requiere una organización gestionada y la participación de todo el personal que trabaja con el sistema de información. Estos sistemas de información rara vez son inmutables, ya que, se ven sometidos a evolución continua tanto propia (nuevos activos) como externa (nuevas amenazas), lo que hace que sea necesario una revisión periódica adaptada al nuevo contexto.

El análisis de riesgos proporciona un modelo del sistema en términos de activos, amenazas y salvaguardas, siendo fundamental para controlar todas las actividades. Este análisis de riesgos se encuentra encuadrado dentro de la fase de planificación del denominado ciclo PDCA y establecido por la ISO 27000. En la figura 3.4 se puede observar este ciclo.



Figura 3.4 Ciclo PDCA [8]

En resumen, MAGERIT proporciona una guía para realizar la gestión de riesgos de una organización, a partir de la cual, se realizará un plan de seguridad que materializará las decisiones adoptadas para el tratamiento de los riesgos. Así, se recomienda incorporar durante la fase de desarrollo del software las funciones y mecanismos que refuerzan la seguridad del nuevo sistema y de incluso el propio proceso de desarrollo, ya que, tomar

en consideración la seguridad del sistema antes y durante su desarrollo es más efectivo y económico que integrar la seguridad a posteriori.

3.2 Seguridad en bases de datos

Como se ha comentado a lo largo del TFG, la información cada vez es más un recurso indispensable para todas las organizaciones, y por ello, es necesario que el almacenamiento y organización de la misma se realice de la forma más segura posible.

Una de las formas más extendidas de realizar es mediante el uso de bases de datos relacionales, así no es de extrañar que estas sean el objetivo de ataques para obtener, modificar o destruir la información que contienen. Para evitar estos ataques existen una serie de mecanismos como el control de acceso o la criptografía [22].

3.2.1 Control de acceso

El sistema gestor de bases de datos (SGBD) debería incluir este mecanismo de seguridad, que permite restringir el acceso al sistema como un todo. Para ello, es necesario crear distintas cuentas de usuario y contraseñas, a las cuales, se les asignarán distintos permisos controlando el proceso de entrada al sistema.

Relacionado con este mecanismo de seguridad se encuentra el principio de menor privilegio [23], que afirma que cualquier objeto (usuario, administrador, programa, etc.) debe tener tan solo los privilegios de uso necesarios para desarrollar su tarea y ninguno más.

Este tema adquiere mayor importancia si tenemos en cuenta que según un estudio realizado por el instituto Infosec [24], el coste debido a pérdidas de información provocadas por el denominado factor humano se encuentra en tercera posición tras los ataques criminales maliciosos y los fallos de los sistemas.

Estos ataques que se producen desde dentro de la organización pueden tener distintos orígenes: empleados actuales con ataques intencionados (el usuario desea acceder información para realizar un mal uso de la misma), empleados actuales con ataques no intencionados (el usuario no tiene intención de realizar un ataque, pero su desconocimiento en temas relacionados con la seguridad provoca una amenaza al sistema. Esto se podría solucionar mediante cursos de formación para empleados) y antiguos empleados (en este caso, el usuario ya no debería poder entrar a la información del sistema, pero al no habersele revocado sus permisos puede acceder a la misma. Por ello, es muy importante asegurarse que los usuarios ya no presentes en el sistema carecen de permisos).

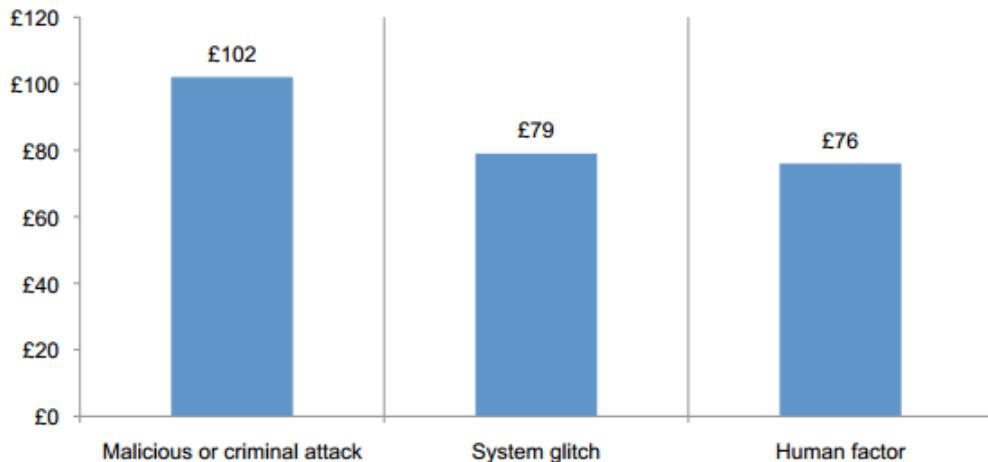


Figura 3.5 Coste per cápita de las principales causas de pérdida de información [24]

3.2.2 Cifrado de datos

Otro mecanismo de seguridad es el cifrado de datos, el cual, sirve para proteger los datos confidenciales transmitidos por cualquier tipo de red de comunicaciones. El cifrado provee de una protección adicional a secciones que deben permanecer confidenciales en la base de datos [22]. En general, los algoritmos criptográficos se pueden clasificar en tres grandes familias [25]:

- **Criptografía simétrica o de clave secreta:** se crea un mensaje cifrado utilizando una única clave conocida por los dos interlocutores, de manera que el documento cifrado sólo pueda descifrarse conociendo dicha clave secreta. Algunos ejemplos de algoritmos de criptografía simétrica son: DES (Data Encryption Standard) o AES (Advanced Encryption Standard) [26].
- **Criptografía asimétrica o de clave pública:** en este caso, se utilizan dos claves distintas para cifrar y descifrar el mensaje. Cada usuario cuenta con una pareja de claves: una privada y una pública. Así para mandar a un usuario un mensaje se cifrará con su clave pública, el mensaje sólo podrá ser descifrado con la clave privada del receptor. Un ejemplo de algoritmo de criptografía asimétrica es: RSA [27].
- **Algoritmos HASH:** se tiene una información de entrada con una longitud indeterminada y se obtiene como salida un código que se puede considerar único para cada entrada. Así de una misma entrada siempre se obtendrá la misma salida.

Mediante este mecanismo se consigue que aunque un usuario no autorizado consiga tener acceso a los datos del sistema, al encontrarse codificados tendrá problemas para descifrarlos.

3.3 Metodología para el diseño de bases de datos seguras

Como se ha comentado anteriormente, la mejor forma para integrar la seguridad en un sistema, es que, todo el ciclo de vida del sistema tenga en cuenta la seguridad, no tratarlo como un añadido. Por ello, para finalizar este estado del arte se comentará una metodología propuesta para realizar el diseño de bases de datos de forma segura. Esta metodología forma parte de la tesis doctoral de Eduardo Fernández-Medina Patón [28].

Para ello, la metodología se divide en las tres etapas tradicionales para el diseño de las bases de datos:

- **Modelado conceptual:** en esta etapa se obtendrá una representación adecuada de la información que se desea modelar, utilizando un modelo conceptual (normalmente entidad-interrelación). Esta etapa se dividirá a su vez en distintas etapas como:
 - **Captura de requisitos:** se realiza un análisis de requisitos del sistema, construyendo un glosario de términos, eliminando ambigüedades y agrupando requisitos, prestando especial atención a los requisitos de seguridad.
 - **Análisis del sistema:** su objetivo es formalizar los requisitos capturados en la etapa anterior consiguiendo una representación de los mismos mediante un diagrama de clases. En esta etapa, se refinan los conceptos basándose en los requisitos del sistema. También se integran varios subesquemas en un esquema general, y se verifica que el modelo obtenido es correcto.
- **Diseño lógico:** se transforma el modelo conceptual obtenido de la anterior etapa a un paradigma de modelos lógicos para bases de datos actuales. Esta etapa se dividirá a su vez en distintas etapas como:
 - **Diseño lógico relacional multinivel:** en esta etapa se pretende conseguir un modelo multinivel general a través de transformaciones del diagrama de clases obtenido en la etapa anterior.
 - **Diseño lógico específico:** se dan unas pautas generales a la hora de implantar la base de datos en cualquier sistema de gestión de bases de datos.
- **Diseño físico:** tiene como objetivo obtener una implementación eficiente del modelo lógico anterior, considerando una plataforma y producto concreto.

4 Método de Trabajo

En este capítulo se detallarán tanto la metodología de trabajo, como las tecnologías utilizadas para el desarrollo del proyecto.

Para la gestión del proyecto se utilizará la metodología ágil Scrum [29]. Se ha elegido esta metodología debido a su gran flexibilidad y a su capacidad de adaptación a los cambios de las circunstancias relacionadas con el proyecto; además al tratarse de una metodología basada en incrementos encaja perfectamente con el objetivo de desarrollar las distintas evaluaciones a realizar por la herramienta.

Por otro lado, para la gestión del desarrollo del software se utilizará TDD [30] o desarrollo basado en pruebas. Se ha elegido esta metodología de desarrollo como forma de garantizar la calidad del software desarrollado, así como para ir adaptando el diseño de la solución del problema mientras se va consiguiendo un mayor entendimiento del mismo.

4.1 Scrum

Normalmente se usa Scrum para gestionar el desarrollo de software y productos complejos, para ello hace uso de prácticas iterativas e incrementales. Con Scrum se consigue un aumento en la productividad a la par que reducimos el tiempo necesario para obtener beneficios (en comparación con la típica metodología de desarrollo en cascada).

Los procesos de Scrum permiten a las organizaciones reajustarse rápidamente a los cambios que se produzcan en los requisitos, alcanzando un producto que cumple los objetivos marcados.

Para ello, se basa en la teoría de control de procesos empírica, lo cual asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control de riesgo.

Tres pilares soportan toda la implementación del control de procesos empíricos:

- **Transparencia:** los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo.
- **Inspección:** Los usuarios de Scrum deben inspeccionar frecuentemente los

artefactos de Scrum y el progreso hacia un objetivo para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

- **Adaptación:** Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será accesible, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

4.1.1 Roles

Scrum propone tres roles diferenciados que deben formalizarse: *Propietario del Producto* o *Product Owner*, *Scrum Master* y el *Equipo de Desarrollo*. Además de estos tres roles, se puede hablar también de un cuarto grupo ajeno al desarrollo del producto que serían los *Interesados*.

En los siguientes subapartados, se detallan cada uno de los miembros del equipo de Scrum [31].

4.1.1.1 Propietario del Producto

El Propietario del Producto es la única persona autorizada para decidir sobre cuáles funcionalidades y características funcionales tendrá el producto. Es quién representa al cliente, usuarios del software y todas aquellas partes interesadas en el producto.

Sus principales responsabilidades son:

- Preocuparse por las necesidades del negocio, sabiendo comunicárselas al equipo de Scrum.
- Maximizar el valor para el negocio, es decir, garantizar el retorno de la inversión del proyecto.
- Revisar las funcionalidades del producto, analizando las mejoras que puedan otorgar un mayor valor de negocio.

4.1.1.2 Scrum Master

El Scrum Master es el alma mater de Scrum. Un error común es denominarlo “líder”, pero el Scrum Master no es un líder al uso, sino que es un servidor neutral, que será el

encargado de fomentar e instruir sobre los principios ágiles de Scrum.

Sus principales funcionalidades son:

- Principalmente, garantizar la correcta aplicación de Scrum.
- Resolver los conflictos que entorpezcan el progreso del proyecto.
- Incentivar y motivar al equipo, creando un clima de trabajo colaborativo, fomentar la auto-gestión del equipo e impedir la intervención de terceros en la gestión del equipo.
- Garantizar el cumplimiento de roles y formas de modelo.

4.1.1.3 Equipo de Desarrollo

El equipo de desarrollo es el responsable de desarrollar el producto. Se trata de un grupo formado por personas de diferentes disciplinas: programadores, arquitectos, diseñadores, probadores (*testers*), etc. Además se tratan de grupos auto-organizados, ya que, ellos mismos eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas ajenas al equipo.

4.1.1.4 Interesados

El grupo de interesados sería cualquier persona no reflejada en los roles anteriores pero que, de alguna forma, se interesa en el proyecto, como por ejemplo clientes, usuarios, dirección comercial, etc.

No tienen una función concreta, pero pueden ser de utilidad al proyecto a la hora de asesorar o de realizar sugerencias.

4.1.2 Sprint

En Scrum el trabajo se parte en ciclos regulares de duración denominados sprints o iteraciones. Su duración depende en gran parte del Scrum Master del proyecto, pero suele tener una duración aproximada de una o dos semanas.

Durante cada sprint, se desarrolla una parte del producto y al final de cada sprint se obtiene un resultado completo, un incremento del producto que sea potencialmente entregable [32].

4.1.3 Componentes de Scrum

Scrum está organizado en distintos componentes para facilitar su labor que serán explicados a continuación. En la imagen 4.1 se puede ver un resumen de estos componentes.

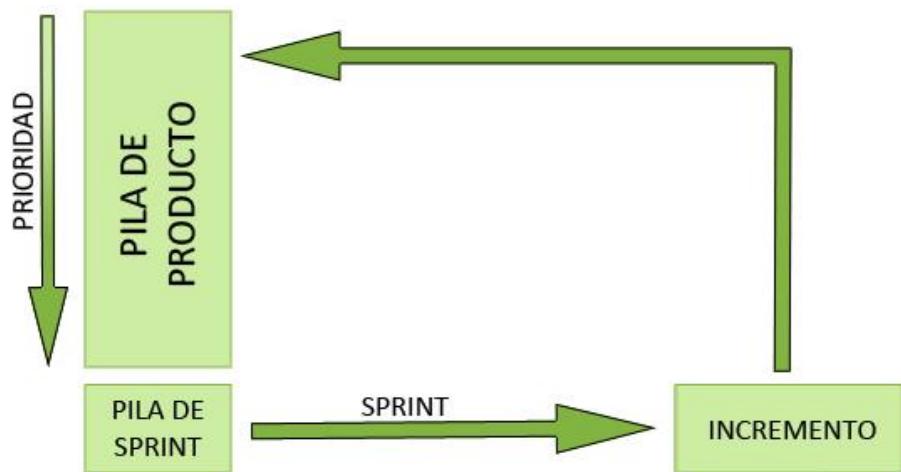


Figura 4.1 Componentes de Scrum

4.1.3.1 *Pila de producto*

La pila de producto es una lista priorizada y evolutiva de funcionalidades de negocio y técnicas que deben ser implementadas por el sistema. Constituye una lista de todos los cambios que deben ser realizados en el producto en futuras actualizaciones.

Se genera cogiendo como origen diversas fuentes, la más normal son los propios requisitos del cliente, pero también pueden venir de otras fuentes como marketing y ventas [33].

4.1.3.2 *Pila de sprint*

Es una lista de trabajos identificados por el equipo de desarrollo que deben ser completados durante un sprint. De esta forma se dividen las funcionalidades listadas de la pila de producto en los distintos sprints que acabaran teniendo como resultado el producto final [34].

4.1.3.3 *Incremento*

Un incremento es una nueva versión del software que sea funcional, con un conjunto de mejoras introducidas en el último sprint [35].

4.2 Desarrollo basado en pruebas (TDD)

TDD [30] es una forma de desarrollar software que combina la realización de un primer test que explica lo que se quiere conseguir con ese test y refactorización. Sus principales objetivos son la especificación y la no validación, es decir, es una forma de desarrollar a partir de los requisitos o del diseño antes de escribir código funcional.

4.2.1 Ciclo de desarrollo de TDD

Para realizar un TDD de una forma correcta será necesario llevar a cabo un ciclo que se compone de los siguientes pasos [36]. En la figura 4.2 se puede ver un esquema que explica su funcionamiento.

1. Elegir un requisito a desarrollar.
2. Escribir un test para dicho requisito.
3. Intentar ejecutar el test. Deberá fallar, ya que, no se cumple la funcionalidad deseada.
4. Crear código específico que resuelva el test.
5. Ejecutar de nuevo el test. En este caso pasará el test, en caso contrario, habrá que modificar el código escrito.
6. En caso de pasar, refactorizar el código para mejorarlo.

4.2.2 Ventajas de usar TDD

Las ventajas de usar TDD frente al enfoque tradicional de desarrollo de software son muy variadas, en la siguiente lista se pueden ver unos cuantos ejemplos de dichas ventajas [37].

- Escribir los tests antes de escribir código hace que realmente consideres quéquieres obtener del código.
- Obtienes información sobre cómo implementarlo de forma rápida.
- Crear una especificación detallada.
- Permite evolucionar al diseño y adaptarlo a los cambios provocados por el

entendimiento del problema.

- Simplificación del código, te ayuda a centrarte en lo importante sin escribir código innecesario.
- Mejora la calidad y reduce los bugs del software a desarrollar.

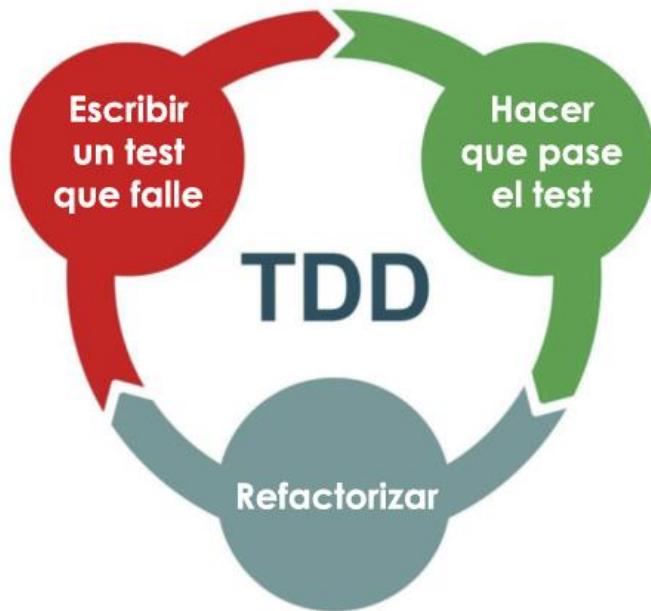


Figura 4.2 Esquema de funcionamiento de TDD

4.3 Kanban

Kanban [38] es un framework usado para implementar metodologías ágiles, fue creado por Toyota y se utiliza para controlar el avance del trabajo, en el contexto de una línea de producción. Kanban basa su estrategia en la mejora continua y continuada. Su objetivo es gestionar de manera general como se van completando las tareas.

Las principales reglas de Kanban son:

1. Visualizar el trabajo y las fases del ciclo de producción o flujo de trabajo.
2. Determinar el límite de trabajo en curso.
3. Medir el tiempo en completar una tarea.

En el caso del TFG que nos ocupa, se utilizará la técnica Kanban a la hora de visualizar las distintas historias de usuario que componen el sistema en tres columnas distintas: Todo, Doing y Done. En la figura 4.3 se puede ver un ejemplo de tablero Kanban.

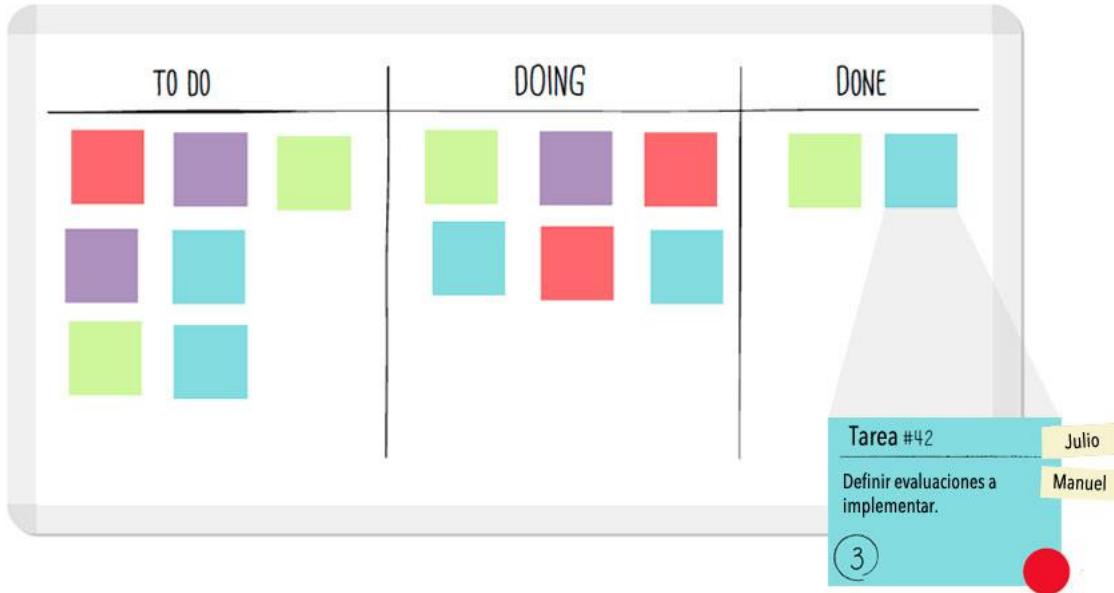


Figura 4.3 Tablero típico de Kanban

4.4 Marco tecnológico para el desarrollo del proyecto

En esta sección se exponen las herramientas y tecnologías utilizadas durante todo el desarrollo del TFG, desde los diagramas previos para comprender mejor el sistema hasta las herramientas y tecnologías necesarias para su implementación.

4.4.1 Herramientas de diseño

4.4.1.1 Visual Paradigm

Visual Paradigm [39] es una herramienta UML Case (Computer-Aided Software Engineering) para el modelado de sistemas que soporta UML 2, SysML y Business Process Modeling Notation (BPMN) de la Object Management Group (OMG). Además del soporte, también tiene funciones para generar informes y para la generación de código.

Para este proyecto se ha utilizado esta herramienta a la hora de realizar cualquier tipo de diagrama UML como, por ejemplo, diagramas de caso de uso, diagramas de secuencias, etc.

4.4.1.2 *Balsamiq Mockups*

Balsamiq Mockups [40] es una aplicación cuyo objetivo es facilitar el desarrollo del diseño de bocetos. Su principal virtud son la sencillez y rapidez con la cual puedes realizar un primer esquema de cómo sería tu aplicación.

Para ello, permite elegir el soporte sobre el que se realizará el boceto (como un Smartphone o una Tablet) al que se le añaden los distintos elementos que acaban formando el boceto final.

Esta aplicación será usada durante este TFG a la hora de realizar los bocetos que representen las distintas interfaces que formen la aplicación web de la herramienta.

4.4.1.3 *Gantt Project*

Gantt Project [41] es una herramienta para para la realización de diagramas de Gantt, los cuales se utilizarán durante el proyecto para expresar de forma gráfica la duración y orden de los distintos sprints.



Figura 4.4 Herramientas de diseño

4.4.2 Herramientas de gestión del proyecto

4.4.2.1 *Trello*

Trello [18-19] es una herramienta de gestión de proyectos que hace que la colaboración sea sencilla. Para ello, se hace uso de tableros (*boards*) que a su vez contienen listas de modo que de un vistazo puedes observar todo lo que hay en tu proyecto. Los ítems dentro de las listas, llamados *cards*, pueden arrastrarse y soltarse en otras listas o reordenarse.

Esta herramienta se utilizará durante el TFG como forma de comunicación entre el tutor

y el autor del proyecto, además de cómo forma de representación de la técnica Kanban, descrita en el apartado 4.3.

4.4.2.2 Hatjitsu

Hatjitsu [44] es una aplicación web que permite realizar planning póker entre los distintos miembros del equipo de Scrum. Para ello, tiene habitaciones a las cuales puedes unirte o crear una nueva para tu equipo.

En este TFG se ha usado Hatjitsu para estimar la duración de los sprints que conforman el proyecto.



Figura 4.5 Herramientas de gestión del proyecto

4.4.3 Herramientas, tecnologías y frameworks para el desarrollo software

4.4.3.1 Apache Hadoop

Apache Hadoop [45] es un *framework* que permite realizar un procesamiento distribuido de grandes cantidades de datos a través de distintos *clusters*, utilizando modelos simples de programación. Está diseñado para ser escalable desde un único servidor a miles de servidores, cada uno ofreciendo computación y almacenamiento local.

Para procesar la gran cantidad de datos que se producen hoy en días, Google inventó un nuevo estilo de procesar datos denominado MapReduce, como forma de aplicar estos conceptos surgió Hadoop [46].

Apache Hadoop incluye un sistema distribuido de ficheros (HDFS), que divide los datos de entrada y los almacena en distintos nodos. Esto hace posible que los datos sean procesados en paralelo haciendo uso de todas las máquinas del *cluster*.

Para el caso de este TFG, Apache Hadoop será el núcleo sobre el que se ejecutarán la mayoría de las evaluaciones que se realizarán sobre la base de datos.

4.4.3.2 *Apache Pig*

Apache Pig [47] es una plataforma para el análisis de grandes cantidades de datos utiliza un lenguaje de alto nivel para expresar programas de análisis de dichos datos. El punto fuerte de los programas en Pig es que su estructura permite parallelización, lo que hace posible manejar enormes cantidades de datos.

En la siguiente imagen se puede observar un ejemplo de la arquitectura de Hadoop, en la cual, se muestran algunas herramientas que se usarán durante el desarrollo de este TFG.

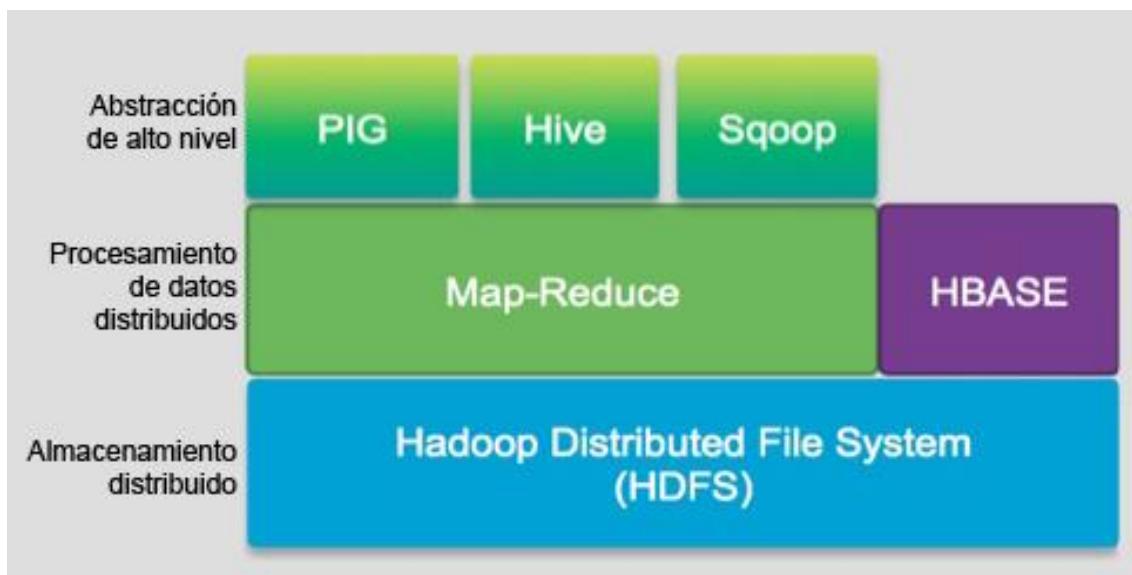


Figura 4.6 Arquitectura de Hadoop

4.4.3.3 *Python*

Python [48] es un lenguaje de programación interpretado cuyo objetivo es hacer una sintaxis que favorezca un código legible. Es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa, e incluso, programación funcional. Se trata de un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Python será utilizado en este proyecto para realizar la implementación de las evaluaciones de seguridad. Como se comentó con anterioridad, la implementación en este proyecto se realizará siguiendo las directivas de la metodología TDD, por tanto, en este caso se usará Nose [49] para la creación de las pruebas.

4.4.3.4 NodeJS

NodeJS [50] es un entorno JavaScript de lado del servidor que utiliza un modelo asíncrono y dirigido por eventos. Para ello, hace uso del motor JavaScript V8 de Google (actualmente uno de los intérpretes más rápidos para cualquier lenguaje dinámico). Uno de los puntos fuertes de NodeJS es su escalabilidad.

En este TFG se utilizará NodeJS para desarrollar el *backend* de la aplicación web que realizará las distintas evaluaciones.

4.4.3.5 AngularJS

AngularJS [51] es un entorno de desarrollo web destinado a crear aplicaciones web de forma simple y fáciles de mantener. Desarrollado por Google, implementa el patrón MVC. AngularJS adapta y amplia HTML para servir mejor contenido dinámico.

Se utilizará para desarrollar el *frontend* de la aplicación web.

4.4.3.6 JSON

JSON [52] es un formato ligero para el intercambio de datos que es fácil de leer y escribir para los humanos, y fácil de analizar y generar para las máquinas. Está basado en un subconjunto del lenguaje de programación JavaScript. JSON es un formato de texto que es totalmente independiente de cualquier lenguaje de programación, pero que usa convenciones que les son familiares a los programadores.

En este proyecto será utilizado este tipo de mensajes para comunicar el sistema *backend* (NodeJS) con el sistema *frontend* (AngularJS).

4.4.3.7 *Bootstrap*

Bootstrap [28-29] es un entorno de desarrollo web, creado por Twitter, que permite crear interfaces web con CSS y Javascript que adaptan la interfaz dependiendo del tamaño del dispositivo en el que se visualice de forma nativa, es decir, automáticamente se adapta el tamaño del dispositivo, sin necesidad de que el usuario tenga que hacer nada (*Responsive Design*).

Bootstrap será utilizado en este TFG para ayudar al desarrollo de las interfaces de la herramienta.

4.4.3.8 *HTML*

HTML [30-31] es un lenguaje de marcas que hace posible presentar información en Internet. Lo que se visualiza en una página web es la interpretación que hace el navegador del código HTML. Para ello tiene una estructura básica, basando su funcionalidad en el uso de etiquetas para especificar las distintas partes que componen la página.

4.4.3.9 *CSS*

CSS (*Cascading Style Sheets*, Hojas de Estilo en Cascada) [57], es un mecanismo simple que describe cómo se va a mostrar un documento en pantalla, o como se va a imprimir, o incluso como va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores control sobre el estilo y formato de sus documentos.

4.4.3.10 *SublimeText 3*

SublimeText [33-34] es un sofisticado editor de texto especialmente diseñado para código de programación. Dispone de una gran cantidad de funcionalidades útiles y cómodas desde el punto de vista de la usabilidad y eficiencia, convirtiendo el trabajo de edición de texto en una experiencia sencilla y agradable.

SublimeText será utilizado durante este TFG a la hora de escribir el código necesario para realizar las evaluaciones de seguridad.

4.4.3.11 Google Charts

Google Charts [60] es una extensión utilizada para visualizar datos en una página web de una forma sencilla. Desde simples diagramas de líneas hasta complejos diagramas de árbol. Para conseguirlo se hace uso de código JavaScript embebido en la página web.

Para el caso que nos ocupa, se usará Google Charts para representar los resultados obtenidos de realizar las evaluaciones; los datos se visualizarán mediante dos tipos de diagramas: de sectores y de barras.



Figura 4.7 Herramientas de desarrollo

4.4.4 Herramientas para la gestión de bases de datos

4.4.4.1 MySQL

MySQL [61] es el sistema de gestión de bases de datos (SGBD) de código abierto más popular, está desarrollado por Oracle Corporation. Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL usa bases de datos relacionales que almacenan la información en tablas separadas en

lugar de guardar todos los datos en un solo almacén.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su integración con los lenguajes de programación más utilizados como Java y PHP. También permite su integración con la mayoría de sistemas operativos.

El hecho de ser un software de código abierto ha favorecido su desarrollo y continuo desarrollo, siendo actualmente una de las herramientas más utilizadas por los programadores orientados a Internet [62].

El objetivo principal del proyecto es la evaluación de la seguridad de bases de datos MySQL, por tanto, se trata de una parte muy importante del mismo.

4.4.4.2 Apache Sqoop

Apache Sqoop [63] es una herramienta diseñada para realizar eficientemente la transferencia de todos los datos entre Apache Hadoop y almacenes de datos estructurados, como bases de datos relacionales.

Apache Sqoop será utilizado en este TFG para convertir las tablas de la base de datos a evaluar a un formato legible para Apache Hadoop.

4.4.4.3 MongoDB

MongoDB [64] es una base de datos potente, flexible y escalable que combina su habilidad de escalar con otras funcionalidades como índices secundarios, gran variedad de consultas, ordenamiento y agregación.

MongoDB no es una base de datos relacional, sino que se basa en el concepto de documento, gracias al cual el modelo de columnas se vuelve mucho más flexible, favoreciendo la escalabilidad. Además no se encuentra limitado a esquemas predefinidos como tipos o tamaños, lo que hace que sea más sencillo añadir o quitar campos.

En este TFG se utilizará esta tecnología a la hora de almacenar la información sobre las distintas evaluaciones, ya que, puede ser necesario cambiar el formato de almacenamiento de las mismas, lo cual será más sencillo con MongoDB. Para ello se hará uso de la herramienta MongoLab, el cual, nos permite utilizar MongoDB como un servicio.



Figura 4.7 Herramientas para la gestión de bases de datos

4.4.5 Herramientas para realizar la documentación

4.4.5.1 Microsoft Word

Microsoft Word [65] es un procesador de texto desarrollado por Microsoft perteneciente a la suite de aplicaciones ofimáticas de Microsoft Office.

Esta herramienta se ha utilizado a la hora de redactar la memoria del TFG.

4.4.5.2 Zotero

Zotero [66] es un programa de software libre para la gestión de referencias bibliográficas. Zotero permite a los usuarios recolectar, administrar y citar investigaciones de todo tipo, para ello importa datos directamente desde las páginas web visualizadas en el momento.

En este TFG se ha utilizado para la recogida y organización de las referencias utilizadas para su creación, para ello se ha utilizado el plugin disponible para Firefox y compatible con Microsoft Word.

4.4.5.3 GIMP

GIMP (GNU Image Manipulation Program) [67] es una herramienta de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Se trata de un programa libre y gratuito.

Se utilizará a la hora de modificar y crear las imágenes usadas en la memoria del TFG.



Figura 4.8 Herramientas para realizar la documentación

4.4.6 Herramienta para el uso de máquinas virtuales

4.4.6.1 *VirtualBox*

VirtualBox [68] es un software de virtualización para arquitecturas x86/amd64. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como “sistemas invitados”, dentro de otro sistema operativo “anfitrión”, teniendo cada uno su propio ambiente virtual.

VirtualBox se ha utilizado en este TFG a la hora de crear una máquina virtual con sistema operativo Lubuntu, donde se instalarán los componentes necesarios para realizar las evaluaciones de seguridad implementadas.



Figura 4.9 Herramienta para el uso de máquinas virtuales

Finalmente en la siguiente imagen se puede ver una recopilación de las distintas herramientas que se utilizarán durante el desarrollo del TFG, agrupadas según su funcionalidad.



Figura 4.10 Resumen de herramientas usadas en el proyecto

5 Resultados

En este capítulo se explicarán los resultados obtenidos tras aplicar las distintas metodologías definidas en la sección 4. Para ello, se estructurará el capítulo en función de los sprints que forman la metodología Scrum.

Así pues, en un primer momento será necesario llevar a cabo un sprint 0 inicial que sirva para identificar las distintas evaluaciones a realizar durante el proyecto, así como los distintos sprints y gestiones del proyecto. En cada apartado, se explicarán con detalle todos los pasos necesarios para obtener el incremento final esperado de cada sprint.

5.1 Sprint 0

El Sprint 0 tiene como objetivo principal la definición del alcance del proyecto, así como la planificación del mismo, definiendo la pila de producto y de sprints que servirán de base al resto del proyecto.

Estas definiciones incluirán cuestiones como la estimación temporal de cada sprint, su coste, los riesgos del proyecto y como se gestionarán los recursos tanto humanos como técnicos.

5.1.1 Gestión de recursos humanos

Como se explicó en el apartado 4.1.1, los roles asociados a la metodología Scrum son: propietario del producto, Scrum master, equipo de desarrollo e interesados. En este proyecto, los roles del equipo Scrum se dividirán de la siguiente forma:

- Propietario del producto: tutor del TFG.
- Scrum master: autor del TFG.
- Equipo de desarrollo: autor del TFG.
- Interesados: evaluadores que usen la herramienta.

Como se puede ver, al tratarse de un proyecto pequeño realizado por una sola persona, esta adquiere las responsabilidades tanto de Scrum master como de equipo de desarrollo.

Por otro lado, los interesados no se encuentran definidos, ya que, serían aquellas personas que hagan uso de la herramienta, en este caso los evaluadores, los cuales son sus clientes potenciales.

5.1.2 Alcance del proyecto

En este subapartado, se definirán las distintas evaluaciones a realizar sobre la base de datos, las cuales son el núcleo de la herramienta a desarrollar. Para decidir cuáles evaluaciones se iban a realizar se realizó una lista con las posibles evaluaciones que se podían realizar, tras estudiar el estado de la cuestión del capítulo 3, tras ello se seleccionaron las que se consideraron más interesantes. En este punto, es necesario recalcar que se trata de una herramienta que busca tener una gran flexibilidad, es decir, será posible añadir nuevas evaluaciones al sistema de forma sencilla. En la tabla 5.1 se pueden ver las evaluaciones elegidas.

Id	Evaluación	Objetivo	Motivación
EV 1	Evaluar cifrado de columnas de una tabla de una base de datos.	El objetivo de esta evaluación es comprobar que los registros de una columna se encuentran cifrados.	Uno de los mecanismos de seguridad de las bases de datos consiste en el cifrado de la información relevante.
EV 2	Evaluar permisos de usuarios.	El objetivo de esta evaluación es comprobar qué usuarios cumplen con los permisos que se les suponen.	Para evaluar el cumplimiento del principio de menor privilegio, explicado en el capítulo 3.
EV 3	Evaluar número máximo de accesos por hora de una lista de usuarios.	El objetivo de esta evaluación es comprobar que un conjunto de usuarios cumplen con el máximo de accesos por hora que se les suponen.	Los propios usuarios suponen un gran porcentaje de los ataques que se realizan sobre los sistemas.
EV 4	Evaluar que usuarios eliminados del sistema carecen de privilegios.	El objetivo de esta evaluación es comprobar que usuarios que ya no pertenecen al sistema han sido eliminados.	La ISO 27002 en su apartado dedicado a los recursos humanos insta a proteger los intereses de la organización cuando los usuarios ya no pertenecen al sistema.
EV 5	Evaluar que una lista de usuarios sólo accede en sus horas de trabajo.	El objetivo de esta evaluación es comprobar que una lista de usuarios accede al sistema sólo durante las horas que tiene permitido.	Al igual que en la evaluación 3, es necesario tener un control sobre los accesos de los usuarios.
EV 6	Evaluar la cantidad de intentos de accesos fallidos.	El objetivo de esta evaluación es comprobar qué cantidad de accesos fallidos se han producido y con qué usuario.	Repetidos accesos fallidos sobre un mismo usuario puede indicar un intento de ataque al sistema.

Tabla 5.1 Evaluaciones a realizar

5.1.3 Plan de proyecto

Teniendo las evaluaciones que se van a implementar en este TFG se puede crear la pila de producto, que estará formada por las historias de usuario del sistema, su valor de negocio y una estimación del tiempo necesario para realizarla. Es necesario destacar que para saber el valor de negocio de cada historia de usuario se ha consultado con el propietario del producto.

Por otro lado, para la estimación temporal también se ha contado con la opinión del propietario del producto, para ello se ha utilizado la técnica del planning poker y la herramienta Hatjitsu (explicada en el apartado 4.4.2.2), como curiosidad cabe destacar que para la estimación se usaron como números posibles los de la serie de Fibonacci. Al realizarse entre dos personas el planning poker se realizaba siguiendo los siguientes pasos:

1. Cada uno elegía el tiempo que estimaba que se tardaría en realizar dicha historia de usuario.
2. En caso de coincidir, se elegía esa estimación.
3. En caso de diferir en la estimación, cada uno explicaba sus motivos y se volvía a repetir la estimación.
4. Si se volvían a dar resultados dispares se solucionaba haciendo la media entre las dos estimaciones.

En la tabla 5.2 se representa la pila de producto priorizada resultante de aplicar estos procedimientos. Se han añadido historias adicionales que son necesarias para la realización de dichas evaluaciones, además de otras historias de usuario para indicar la intención del cliente de que debe ser una herramienta web.

Identificador	Historia de usuario	Estimación en horas	Valor de negocio
HdU 1	Quiero poder añadir y procesar una base de datos para su evaluación.	34 horas	Alto
HdU 2	Quiero comprobar que los datos de una columna se encuentran cifrados.	21 horas	Alto
HdU 3	Quiero comprobar qué usuarios tienen un cierto privilegio.	27,5 horas	Alto
HdU 4	Quiero comprobar que un usuario sólo puede realizar un número máximo de accesos.	21 horas	Alto
HdU 5	Quiero comprobar que los usuarios que ya no pertenecen al sistema ya no disponen de permisos.	27,5 horas	Alto

HdU 6	Quiero evaluar que una serie de usuarios acceden sólo durante las horas de trabajo.	27,5 horas	Alto
HdU 7	Quiero evaluar la cantidad de accesos fallidos realizados sobre la base de datos.	27,5 horas	Alto
HdU 8	Quiero obtener unos gráficos para representar el informe de resultados.	10,5 horas	Alto
HdU 9	Quiero poder introducir un archivo con el log de la base de datos en el sistema.	5 horas	Medio
HdU 10	Quiero que sea una aplicación web.	34 horas	Medio
HdU 11	Quiero que la herramienta tenga una interfaz.	17 horas	Medio
HdU 12	Quiero tener un diccionario en el sistema para comprobar si se encuentran cifrados los campos.	10,5 horas	Bajo
HdU 13	Quiero introducir un archivo con los usuarios eliminados del sistema.	3 horas	Bajo
HdU 14	Quiero introducir un archivo con los usuarios a evaluar y su horario de trabajo.	5 horas	Bajo
HdU 15	Quiero obtener un pdf del informe con los resultados de la evaluación de seguridad.	10,5 horas	Bajo

Tabla 5.2 Pila de producto priorizada

Una vez tenemos la pila de producto priorizada definida, se puede dividir en los distintos sprints que tendrán como resultado final la herramienta objetivo del TFG. En este caso, se ha decidido dividir las historias de usuario por la evaluación que ayudan a crear, es decir por su funcionalidad, más que por la paridad en el tiempo estimado para su realización. Con esto, se consigue que al final de cada sprint se consiga una pequeña parte funcional e independiente del sistema.

Se ha decidido realizar primero el sprint necesario para poder introducir la información de la base de datos en el sistema Big Data, ya que, será necesario para la mayoría de las evaluaciones.

Tras esto se realizarán los sprints correspondientes a las evaluaciones de seguridad, y finalmente se tendrán dos sprints: uno para implementar el servidor web que sirve de interfaz entre el cliente y el sistema y otro para la generación del informe final.

En la tabla 5.3 y 5.4 se puede observar cómo se han dividido las historias de usuario en sprints, qué artefactos y validaciones se pretenden conseguir al final de cada sprint.

Sprint	Historias de usuario		Estimación	Artefactos	
1	HdU 1		34 horas	<ul style="list-style-type: none"> • Prototipo de cómo usar la base de datos en conjunto con la tecnología Big Data. 	
2	HdU 2	HdU 12	31,5 horas	<ul style="list-style-type: none"> • Prototipo que implemente la regla de evaluación EV 1. • Diccionario con todas las palabras en castellano. 	
3	HdU 3		27,5 horas	<ul style="list-style-type: none"> • Prototipo que implemente la regla de evaluación EV 2. 	
4	HdU 4		21 horas	<ul style="list-style-type: none"> • Prototipo que implemente la regla de evaluación EV 3. 	
5	HdU 5	HdU 13	30,5 horas	<ul style="list-style-type: none"> • Prototipo que implemente la regla de evaluación EV 4. 	
6	HdU 6	HdU 9	HdU 14	37,5 horas	<ul style="list-style-type: none"> • Prototipo que implemente la regla de evaluación EV 5. • Archivo de log en el sistema.
7	HdU 7		27,5 horas	<ul style="list-style-type: none"> • Prototipo que implemente la regla de evaluación EV 6. 	
8	HdU 10	HdU 11		51 horas	<ul style="list-style-type: none"> • Servidor. • Comunicación del servidor con todas las evaluaciones a realizar. • Interfaz web.
9	HdU 8	HdU 15		21 horas	<ul style="list-style-type: none"> • Prototipo de informe final representado mediante gráficos. • Prototipo de informe final representado mediante un pdf.

Tabla 5.3 Plan de proyecto detallado

Sprint	Historia de usuario	Estimación en horas	Validación
1	Quiero poder añadir una base de datos para su evaluación.	34 horas	Poder añadir al sistema Hadoop una base de datos del cliente .
2	Quiero comprobar que los datos de una columna se encuentran cifrados.	21 horas	Obtener número de registros que cumplen e incumplen esta premisa.
	Quiero tener un diccionario en el sistema para comprobar si se encuentran cifrados los campos.	10,5 horas	Tener un diccionario con todas las palabras para ser consultado.
3	Quiero comprobar qué usuarios tienen un cierto privilegio.	27,5 horas	Obtener cuáles usuarios disponen de este privilegio.
4	Quiero comprobar que un usuario sólo puede realizar un número máximo de conexiones.	21 horas	Obtener si cumple o no esta condición.
5	Quiero comprobar que los usuarios que ya no pertenecen al sistema ya no disponen de permisos.	27,5 horas	Obtener una lista con aquellos permisos que no han sido retirados de usuarios supuestamente eliminados.
	Quiero introducir un archivo con los usuarios a evaluar.	3 horas	Tener un archivo con los usuarios a evaluar.
6	Quiero evaluar que una serie de usuarios acceden sólo durante las horas de trabajo.	27,5 horas	Obtener una lista con aquellos usuarios que han accedido a la base de datos fuera de su horario.
	Quiero poder introducir un archivo con el log de la base de datos en el sistema.	5 horas	Tener un archivo con el log de la base de datos.
	Quiero introducir un archivo con los usuarios a evaluar y su horario de trabajo.	5 horas	Tener un archivo con los usuarios y su horario para evaluarlo.
7	Quiero evaluar la cantidad de accesos fallidos realizados sobre la base de datos.	27,5 horas	Obtener el número de fallos al acceder a la base de datos y sobre qué usuario se produjeron.
8	Quiero que sea una aplicación web.	34 horas	La aplicación resultante es web.
	Quiero que la herramienta tenga una interfaz.	17 horas	Interfaz para la herramienta.
9	Quiero obtener unos gráficos para representar el informe de resultados.	10,5 horas	Gráficos representando los resultados.

	Quiero obtener un pdf del informe con los resultados de la evaluación de seguridad.	10,5 horas	Informe pdf generado con los resultados.
--	---	------------	--

Tabla 5.4 Historias de usuario del sistema

5.1.4 Gestión temporal del proyecto

Una vez definido el alcance del proyecto, con sus sprints e historias de usuario que lo formarán, se creará un diagrama de Gantt para representar de forma gráfica la estimación temporal de desarrollo del proyecto.

Se ha elegido el día 6 de abril como fecha de inicio del primer sprint, es necesario tener en cuenta que antes de empezar con la realización de dicho sprint ha sido necesario hacer el sprint 0. Finalmente, es importante indicar que para hacer este sprint 0 se han dedicado aproximadamente 2 semanas. Otro asunto a considerar, antes de la representar el diagrama de Gantt, es la duración de la jornada laboral; debido al resto de obligaciones del autor del TFG se ha decidido que la duración normal será de unas 4 horas diarias, incluyendo los fines de semana como días en los que el autor podrá seguir trabajando.

Como se indicó en el capítulo 4, el diagrama se realizará mediante la herramienta Gantt Project. Los sprints se han dibujado con otro color para facilitar su identificación. Todas las relaciones entre historias de usuario serán de inicio-fin, ya que, no será posible paralelizar ninguna tarea al tener todas unos objetivos definidos y necesarios para la realización de la siguiente historia de usuario. Además el hecho de que el equipo de desarrollo se encuentre formado por una sola persona dificulta la posibilidad de realizar paralelamente varias historias de usuario.

Es necesario añadir también que en esta estimación temporal no se ha tenido en cuenta la carga de trabajo necesaria para la realización de la memoria del TFG, al no considerarse una necesidad propia del cliente sino un complemento de la herramienta. Esta memoria del TFG se irá completando a medida que se vayan desarrollando los sprints, de forma que no se convierta en un gran esfuerzo final. Aún así, se estima que a la memoria se le dedicará individualmente unas 4 semanas. Finalmente, sólo queda indicar que la estimación temporal final del proyecto son unas 405 horas, sin contar el tiempo necesario para otros asuntos como reuniones con el tutor. En la figura 5.1 se puede ver el diagrama de Gantt resultante de todo lo expuesto con anterioridad.

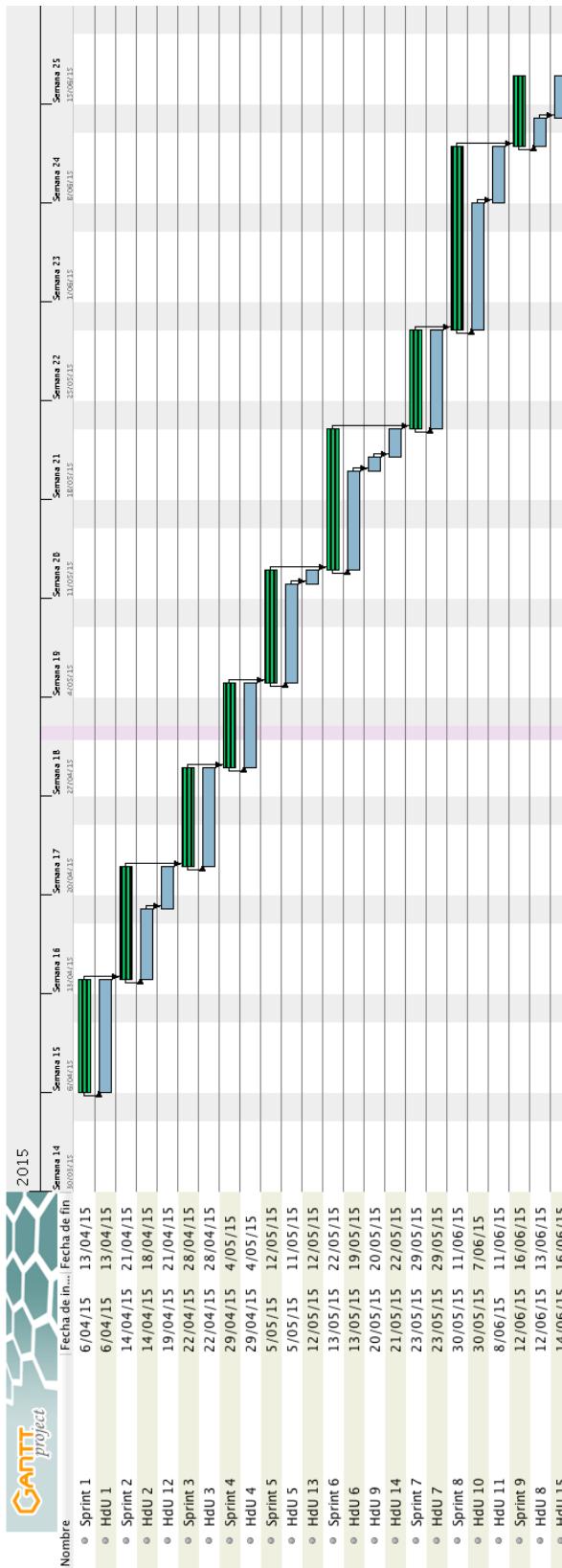


Figura 5.1 Diagrama de Gantt

5.1.5 Gestión de las comunicaciones

Las comunicaciones entre el autor y el director del proyecto para informar sobre el estado del proyecto y resolver las dudas que pudiesen surgir durante su desarrollo, tanto por parte del autor como del director, se realizarán mediante reuniones bajo petición. Esta petición normalmente se realizará mediante correo electrónico. Como mínimo se realizará una reunión semanal.

Además, otra forma de informar al director del estado del proyecto será la utilización de la técnica de gestión Kanban; para aplicar esta técnica se utilizará la herramienta online Trello. Tanto Kanban como Trello se encuentran explicados en el capítulo 4.

Para ello, se creará, y se compartirá con el director de proyecto, un nuevo tablero por cada sprint, además de uno general con todos los sprints. De esta forma, se podrá ver de forma rápida e intuitiva los avances del proyecto. Además será posible comunicarse, ya que, Trello permite realizar comentarios sobre las distintas tareas.

En la figura 5.2 se puede ver un ejemplo del desarrollo de un sprint, mientras que en la figura 5.3 se muestra el tablero general con todos los sprints que componen la herramienta y su estado.



Figura 5.2 Ejemplo de tablero en Trello (Sprint 6)

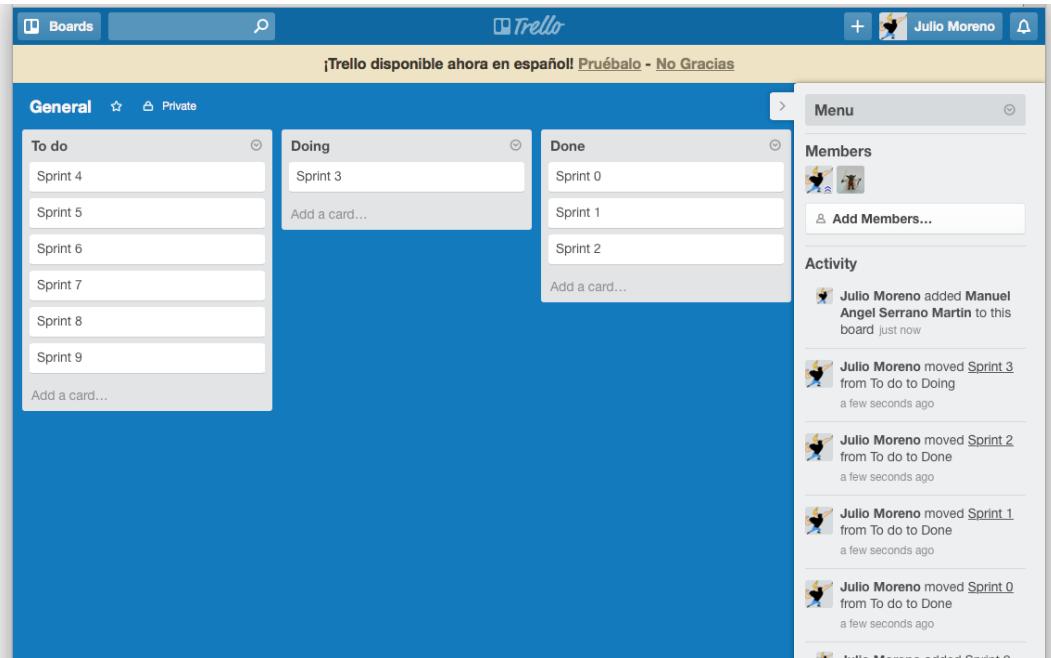


Figura 5.3 Tablero general para gestionar el desarrollo de los sprints

5.1.6 Gestión de recursos

Para el desarrollo del proyecto se ha utilizado un equipo con las siguientes características:

- Procesador Intel Core i5 1,7 GHz
- Memoria RAM de 8 GB
- Windows 8.1 como sistema operativo nativo

Es necesario aclarar que, como se ha comentado con anterioridad, para el desarrollo del proyecto se ha utilizado una máquina virtual con las siguientes características:

- Uso de dos procesadores del ordenador anfitrión.
- Memoria RAM de 5 GB
- Lubuntu 12.04.3 LTS

5.1.7 Gestión de riesgos.

Para la gestión de los riesgos del proyecto se ha decidido utilizar una lista de comprobación con los riesgos más comunes a los que se enfrenta un proyecto durante su desarrollo. La lista utilizada en este caso es la propuesta en [69]. En la tabla 5.5 se

ilustran los posibles riesgos a los que se enfrenta el proyecto agrupados por categorías indicando el impacto estimado en días en el desarrollo y la probabilidad de que sucedan.

Riesgo	Probabilidad	Impacto (en días)
A. Elaboración de la planificación		
A.1. Planificación optimista, “mejor caso”.	20%	5
A.7. El esfuerzo es mayor que el estimado.	30%	7
A.11. Un retraso en una tarea produce retrasos en cascada en las tareas dependientes.	35%	5
B. Organización y gestión		
B.2. El proyecto languidece en el inicio difuso.	10%	3
C. Ambiente/infraestructura de desarrollo		
C.5. Las herramientas de desarrollo no funcionan como se esperaba; el personal de desarrollo necesita tiempo para resolverlo o adaptarse.	30%	5
C.7. La curva de aprendizaje para la nueva herramienta de desarrollo es más larga de lo esperado.	25%	4
D. Usuarios finales (no se aplica)		
E. Cliente		
E.4. El tiempo de comunicación del cliente es más lento de lo esperado.	5%	2
E.5. El cliente insiste en decisiones técnicas que alargan la planificación.	10%	7
F. Personal contratado (no se aplica)		
G. Requisitos		
G.2. Los requisitos no se han definido correctamente y su redefinición aumenta el ámbito del proyecto.	10%	3
G.3. Se añaden requisitos extra.	10%	7
H. Producto		
H.3. Utilizar lo último en informática alarga la planificación de forma impredecible.	15%	5
H.5. El desarrollo de una interfaz de usuario inadecuada requiere volver a diseñarla e implementarla.	10%	7
I. Fuerzas mayores (no se aplica)		
J. Personal		
J.5. La falta de motivación y de moral reduce la productividad.	5%	2
J.7. El personal necesita un tiempo extra para acostumbrarse a trabajar con herramientas o entornos nuevos.	30%	4
J.22. El personal trabaja más lento de lo esperado.	15%	5

K. Diseño e implementación		
K.9. Los componentes desarrollados por separado no se pueden integrar de forma sencilla, teniendo que volver a diseñar y repetir algunos.	30%	7
L. Proceso (no se aplica)		

Tabla 5.5 Análisis de riesgos

Una vez estudiados los riesgos a los que se enfrenta el proyecto, podemos concluir que al tratarse de un proyecto con una sola persona en el equipo de desarrollo, las posibilidades de que se concrete una amenaza son más bajas, además se dispone de un pequeño margen de tiempo para poder hacer frente a algún contratiempo.

5.1.8 Gestión de costes.

Para gestionar los costes derivados del desarrollo del proyecto se ha realizado una tabla (Tabla 5.6), en la cual, se detallan los costes estimados que tendría la realización de la herramienta que nos ocupa.

Concepto	Precio	Subtotal
Equipo de desarrollo	15 €/hora	7050 €
Hardware	600 €	600 €
Software		
• Visual Paradigm	100 €	100 €
• Microsoft Word	539 €	539 €
Electricidad	0,85 €/hora	348,5 €
Otros gastos	100 €	100 €
	Total	8737,5 €

Tabla 5.6 Gestión de costes

Cabe destacar que la mayoría del software utilizado para la realización de TFG es libre y, por tanto, no supone un gasto extra. Además, para el resto de software se dispone de una licencia académica.

Por otro lado, el hardware es proporcionado por el autor del TFG y las horas de trabajo del equipo de desarrollo, que suponen el mayor coste para el proyecto, tampoco serán cobradas, ya que, también las realizará el autor.

5.2 Sprint 1

Durante este sprint se han llevado a cabo las tareas relacionadas con la adición de una base de datos al sistema Hadoop para poder procesarla. Se puede observar en las tablas 5.3 y 5.4 un resumen de lo que se quiere obtener al realizar este sprint.

5.2.1 Refinamiento de la Pila del Producto

La Pila del Producto no ha sufrido modificaciones tras su revisión.

5.2.2 Planificación del Sprint

Para este sprint, disponemos de la historia de usuario 1 (“Quiero poder añadir una base de datos para su evaluación”). El ser la primera historia de usuario del sistema, hace que sea necesario primero realizar una instalación del sistema Hadoop que servirá como base para la realización de las evaluaciones.

Por otro lado, existen varias formas de manejar los datos de una base de datos con Hadoop, por tanto, será necesario tomar una decisión sobre cómo tratar este tema que afectará al resto del proyecto.

Finalmente, la pila de sprint está formada por una única historia de usuario, que es la siguiente:

Historia de Usuario	
Número: 1	
Nombre historia: Añadir y procesar bases de datos	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 34 horas	Sprint asignado: 1
Programador responsable: Julio Moreno García-Nieto	
Descripción: Poder añadir bases de datos al sistema Hadoop para posteriormente poder procesar sus datos mediante trabajos MapReduce.	
Postcondición: La base de datos quedará añadida en el sistema de archivos de Hadoop (HDFS).	
Tareas: <ul style="list-style-type: none">• Instalación del sistema Hadoop.• Decidir cómo guardar los datos de la base de datos.• Procesar los datos.	

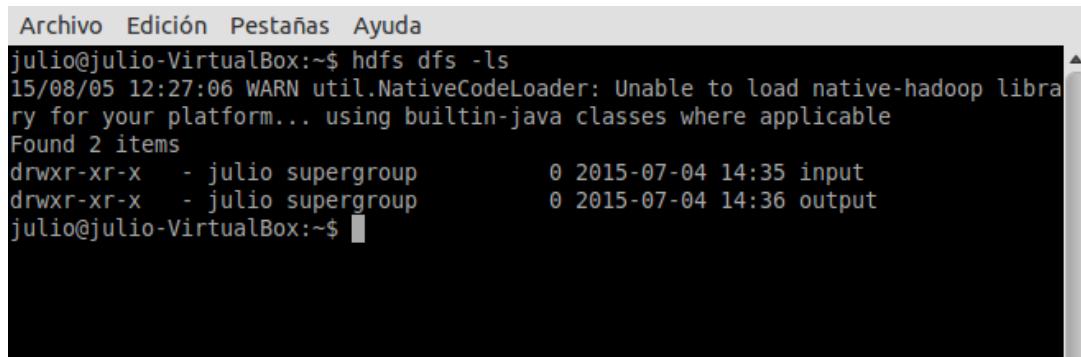
Tabla 5.7 Historia de usuario "Añadir y procesar bases de datos"

5.2.3 Desarrollo de la historia de usuario: Añadir una base de datos para su evaluación

Como se comentó en el anterior apartado, la primera tarea a realizar en esta historia de usuario será la de realizar la instalación del sistema Hadoop. En este caso, se ha decidido realizar la instalación en una única máquina virtual, debido a las limitaciones técnicas del autor del TFG. Dicha máquina virtual contará con un sistema operativo Lubuntu para poder aprovechar al máximo los recursos en las labores de realizar las evaluaciones.

Para ello, se han seguido los siguientes pasos:

1. Instalar Java.
2. Configurar Java.
3. Instalar certificados SSH, necesarios para el nodo maestro.
4. Descargar e instalar Hadoop, se ha instalado la última versión estable que había disponible en el momento, la 2.6.0. Actualmente, se ha lanzado una nueva versión 2.7.1.
5. Configurar Hadoop.
6. Finalmente, se han realizado unas pruebas para comprobar que todo funcionaba como es debido. Por ejemplo, se han creado dos carpetas (input y output) en el sistema de archivos de Hadoop (HDFS) y que serán utilizadas para realizar las distintas evaluaciones.



A screenshot of a terminal window titled "Archivo Edición Pestañas Ayuda". The command entered is "julio@julio-VirtualBox:~\$ hdfs dfs -ls". The output shows a warning message about NativeCodeLoader and then lists two items in the HDFS directory: "input" and "output". Both files were created on 2015-07-04 at 14:35 and 14:36 respectively by user "julio".

```
julio@julio-VirtualBox:~$ hdfs dfs -ls
15/08/05 12:27:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
drwxr-xr-x - julio supergroup          0 2015-07-04 14:35 input
drwxr-xr-x - julio supergroup          0 2015-07-04 14:36 output
julio@julio-VirtualBox:~$
```

Figura 5.4 HDFS funcionando con las carpetas "input" y "output"

7. Como comprobación adicional y para conseguir experiencia programando en esta tecnología, se han realizado algunos ejemplos de MapReduce no relacionados con el tema del TFG.

Una vez tenemos instalado y funcionado el sistema Hadoop, la siguiente tarea a realizar es elegir cómo se va a conseguir utilizar los datos de una base de datos para realizar un trabajo MapReduce.

Existen dos posibilidades en general a la hora de trabajar con Hadoop y bases de datos, utilizar streaming para no tener que almacenar nada en el sistema HDFS, y por otro lado, realizar un volcado de la base de datos en el HDFS de Hadoop para posteriormente utilizar estos datos como entrada del MapReduce.

Cada una de estas posibilidades cuenta con sus ventajas e inconvenientes, la más evidente es el hecho de que haciéndolo mediante streaming no necesitas almacenar datos en el HDFS, pero plantea un problema de recursos en caso de que la base de datos sea demasiado grande.

Esto puede resultar un problema para nuestra herramienta, ya que, no tenemos forma de saber cuál será el tamaño de la base de datos y no queremos limitar el tamaño máximo de entrada. Por ello, se ha decidido utilizar la opción de hacer un volcado de los datos de la base de datos al HDFS.

Para realizar esta opción existen dos posibilidades: Apache Sqoop y MySQL Applier for Hadoop.

- Apache Sqoop [63], como se explicó en el capítulo 4, se trata de una herramienta diseñada para realizar eficientemente la transferencia de todos los datos entre Apache Hadoop y almacenes de datos estructurados, como bases de datos relacionales. Es decir, Sqoop realiza un volcado de los datos de la base de datos al sistema de archivos de Hadoop, creando una copia mediante diferentes archivos.
- MySQL Applier for Hadoop [70], por otro lado, se basa en los logs de MySQL para generar archivos en HDFS, además actualiza en tiempo real las distintas acciones que haya sobre la base de datos, dando la sensación de estar siempre conectado.

Finalmente, una vez estudiadas las dos posibilidades se ha decidido usar Apache Sqoop para realizar el volcado de datos, ya que, no se considera necesario para este TFG que el sistema Hadoop se encuentre siempre conectado con una base de datos como simula MySQL Applier for Hadoop.

Para realizar el volcado de datos de la base de datos se utilizará un comando como el representado en la figura 5.5, este comando ejecutará uno o varios trabajos MapReduce (en función del tamaño de la base de datos) que cogerán los datos y los almacenarán en el HDFS.

```
Archivo Edición Pestañas Ayuda
julio@julio-VirtualBox:~$ sqoop import --connect jdbc:mysql://localhost/datos_madrid
--username root --password root -e 'SELECT descripcion_centro from presupuestos WHERE 1 AND $CONDITIONS' --target-dir /user/julio/input/descripcion -m 1 --direct
```

Figura 5.5 Ejemplo de uso de Sqoop

Es necesario aclarar que para realizar los ejemplos y las pruebas de las evaluaciones, ha sido necesario encontrar una base de datos con un tamaño medianamente grande para que los resultados y los tiempos de ejecución fuesen más cercanos a la realidad. Por ello, se ha elegido una base de datos pública que representa los presupuestos de los que consta la ciudad de Madrid.

Por otro lado, se puede ver en el comando como lo primero necesario es conocer la dirección de la base de datos, así como los datos de acceso de un usuario, luego se ejecuta una consulta de tipo SELECT cuyo resultado será lo que se almacene en uno o varios archivos de HDFS. Existen múltiples opciones en Sqoop para guardar la información, siendo posible también coger todas las tablas de una base de datos.

En la figura 5.6 se puede ver cómo se queda guardado el resultado de la consulta y cuál es el formato en el que deja los datos, por otro lado, cabe destacar que se ha hecho uso de la herramienta Pig por permitir manejar el sistema HDFS de una forma mucho más cómoda.

Figura 5.6 Resultados de ejecutar el comando anterior

5.2.4 Revisión del Sprint

A la finalización del sprint 1, se ha conseguido el primer incremento de la herramienta, ya que, ya es posible tener los datos de una base de datos en el sistema de ficheros HDFS. Por otro lado, cabe destacar que la estimación temporal ha encajado casi perfectamente con la duración real del sprint. En la figura 5.7 se puede ver un resumen de lo conseguido durante el desarrollo del sprint.

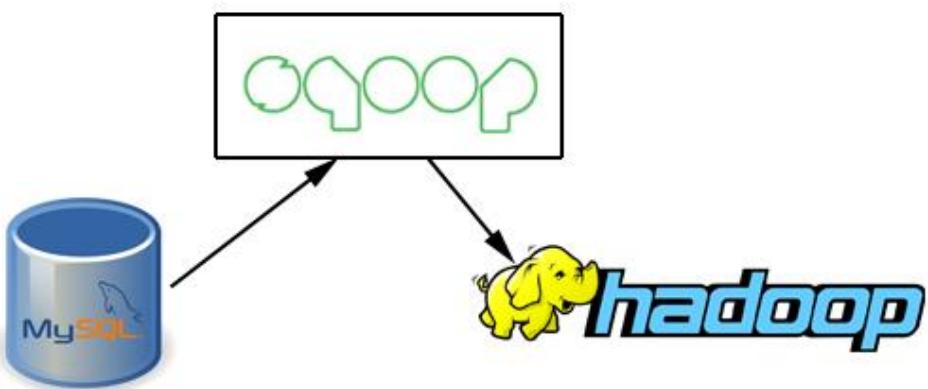


Figura 5.7 Resumen Sprint 1

5.3 Sprint 2

Durante este sprint se llevará a cabo la implementación de la primera evaluación de la herramienta, cuyo objetivo será comprobar la cantidad de registros que se encuentran cifrados dentro de una columna de una base de datos. En las tablas 5.3 y 5.4 se puede ver un resumen de los objetivos de este sprint.

Como se puede observar por la división en historias de usuario, se ha decidido que para la implementación de esta primera evaluación se va a utilizar un diccionario, el cual, servirá de guía para decidir si un término se encuentra cifrado o no, así pues, si dicho término no se encuentra en el diccionario se considerará cifrado.

5.3.1 Refinamiento de la Pila del Producto

El único cambio que se ha producido respecto a la pila del producto original consiste en un cambio en el orden de ejecución de las historias de usuario, ya que, se ha decidido realizar primer la historia de usuario 12 relacionada con el diccionario del sistema, esto es debido a que luego será necesario para implementar la historia de usuario 2 cuyo

objetivo es implementar la evaluación 1 sobre el cifrado de datos.

5.3.2 Planificación del Sprint

Para este sprint, se dispone de dos historias de usuario: la historia de usuario 2 (“Quiero comprobar que los datos de una columna se encuentran cifrados.”) y la historia de usuario 12 (“Quiero tener un diccionario en el sistema.”).

La historia de usuario 2 tiene como objetivo el buscar un diccionario para el sistema, el cual, será alojado en el servidor de la aplicación. Por otro lado, el objetivo de la historia de usuario 12 es el de implementar la primera evaluación del sistema. Para ello, será necesario realizar una serie de pruebas unitarias para luego crear los trabajos MapReduce que realicen la evaluación.

Finalmente, la pila de sprint está formada dos historias de usuario, que son las siguientes:

Historia de Usuario	
Número: 12	
Nombre historia: Tener un diccionario en el sistema	
Valor de negocio: Bajo	Riesgo en desarrollo: Bajo
Esfuerzo: 10,5 horas	Sprint asignado: 2
Programador responsable: Julio Moreno García-Nieto	
Descripción: Añadir un diccionario con todas las palabras en castellano para poder comprobar si un término se encuentra cifrado o no.	
Postcondición: El diccionario quedará añadido en el sistema de archivos de Hadoop (HDFS).	
Tareas: <ul style="list-style-type: none">• Buscar diccionario con todas las palabras.• Convertirlo a fichero de texto.	

Tabla 5.8 Historia de usuario "Tener un diccionario en el sistema"

Historia de Usuario	
Número: 2	
Nombre historia: Comprobar que datos de una columna se encuentran cifrados	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 21 horas	Sprint asignado: 2
Programador responsable: Julio Moreno García-Nieto	
Descripción: Implementar la evaluación 1 del sistema correspondiente con evaluar qué porcentaje de registros de una columna se encuentran cifrados.	

Postcondición:

La evaluación 1 queda implementada.

Tareas:

- Obtener datos de la columna de la base de datos.
- Pruebas unitarias para el mapper.
- Implementar el mapper.
- Pruebas unitarias para el reducer.
- Implementar el reducer.
- Integración de mapper y reducer.

Tabla 5.9 Historia de usuario "Comprobar qué datos de una columna se encuentran cifrados"

5.3.3 Desarrollo de la historia de usuario: *Disponer de diccionario en el sistema*

Como se comentó en el anterior apartado, la primera tarea a realizar en esta historia de usuario será la de buscar un diccionario que encaje con nuestras necesidades. Tras plantearse el problema se ha llegado a la conclusión de que se necesita un diccionario expresado en forma de texto plano.

Para ello, se buscó por internet la solución, encontrándose en [71] una base de datos MySQL con todas las palabras en castellano, lo cual, era útil pero no era exactamente lo que se precisaba. Pero gracias al trabajo realizado en el sprint anterior, se sabía que era posible realizar el volcado de la información de la base de datos a un fichero de texto dentro del HDFS de Hadoop. Para ello, fue necesario hacer de nuevo uso de la herramienta Sqoop.

Para este caso concreto no era necesario crear una consulta, ya que, se necesitaba toda una tabla de la base de datos, por tanto, se ejecutó el comando por consola y, tras ejecutarse el trabajo de MapReduce que Sqoop genera de manera automática, ya se disponía del diccionario del sistema necesario para comprobar si un término se encuentra cifrado.

Pero el resultado de esta operación estaba situado en el HDFS de Hadoop y se necesitaba tener el archivo en local para poder ser usado en el MapReduce y posteriormente situarlo en el servidor. Para realizar esta operación se hizo uso de la herramienta Pig. Cabe destacar que en este archivo resultante se disponía de varias versiones de una misma palabra: una con acentos y otra sin acentos. Con esto, se consigue evitar errores derivados de la introducción de palabras acentuadas o no.

5.3.4 Desarrollo de la historia de usuario: *Comprobar datos de columna se encuentran cifrados*

Como se comentó en la planificación del sprint, la primera tarea a realizar será la de obtener los datos de la columna a evaluar, para ello se utilizará lo aprendido durante el sprint 1.

Una vez tenemos dichos datos, es el momento de empezar a realizar el MapReduce que realice la evaluación. En este punto, considero importante explicar de forma resumida en qué consiste un MapReduce.

MapReduce [72] es un paradigma de programación que permite a los desarrolladores procesar grandes cantidades de datos en paralelo. Para ello, dispone de dos partes:

- El función map toma una serie de valores, los procesa y genera una salida.
- El reducer coge la salida ordenada del mapper como entrada, y genera una salida con valor.

Un ejemplo típico de MapReduce es el de realizar el conteo de palabras que aparecen en un libro. Para realizarlo, se dividirían las páginas en los distintos mappers que componen el sistema, los cuales escribirían cada una de las palabras por separado. Una vez procesadas todas las páginas, se ordenan las palabras salidas de la función map y se le envían a los distintos reducers del sistema, los cuáles se encargarán de sumar 1 cada vez que salga la misma palabra, obteniendo el número de veces que sale cada palabra en el libro. En la figura 5.8 se puede ver un resumen de un proceso MapReduce.

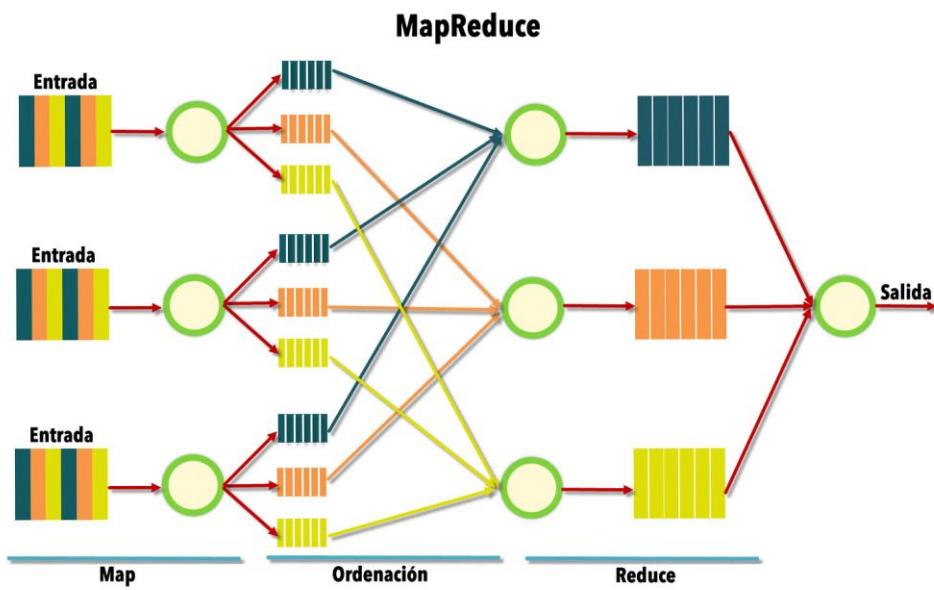


Figura 5.8 Funcionamiento de MapReduce

Una vez se conoce el funcionamiento del MapReduce, se puede iniciar su implementación. En el caso de la primera evaluación, la entrada que se va a tener es la de una columna, cuyos datos no será necesario modificarlos para pasárselos a la función reduce, por tanto el objetivo de la función map en este primer caso será el de recibir cada palabra y volver a escribirla tal cual.

Por otro lado, la función reduce cogerá la salida ordenada de la función map y comprobará si cada término se encuentra en el diccionario o no, en caso de que no se encuentre, se considerará cifrado. Finalmente, se tendrá como salida de la función reduce un porcentaje de registros cifrados o no.

Como se comentó en el capítulo 4, la forma de realizar la programación será usando TDD y en concreto Nose para realizarlas sobre Python, por tanto, lo primero será escribir una serie de pruebas unitarias, las cuales, se pueden observar en el listado 5.1.

```
from nose.tools import assert_equals
from mapper import mapperEV1

def test_mapper():
    assert_equals(mapperEV1("Prueba 1 Mapper"), "Prueba 1 Mapper")
```

Listado 5.1 Pruebas unitarias función map (evaluación 1)

Obviamente, al intentar ejecutar el test fallará, ya que, todavía no se ha implementado nada en la función map. El proceso de creación mediante TDD consistirá en ir escribiendo código poco a poco de forma que emerja a partir de las pruebas escritas.

```
#!/usr/bin/env python

import sys

def mapperEV1(line):
    line = line.strip()

    return line

#!/usr/bin/env python

import sys

def mapperEV1(line):
    for line in sys.stdin:
        line = line.strip()
        print '%s' % (line)
```

Listado 5.2 Función map de la evaluación 1

El código obtenido no se encuentra en el formato que precisa un trabajo de MapReduce, ya que, la entrada deberá ser la estándar, por ello habrá que aplicar una serie de cambios que se considerarán parte de la comentada refactorización de código. Al final, obtenemos un código como el representado en el listado 5.2, en el cual, se puede ver arriba la función obtenida de aplicar TDD y abajo la implementación con los cambios comentados.

Como se puede ver, en este caso, la función map es trivial, ya que, se trata sólo de devolver la misma línea que se recibe. Una vez tenemos la función map funcionando, es hora de crear las pruebas que necesitará pasar la función reduce, las cuales se pueden ver en el listado 5.3. En ellas se comprueba la cantidad de palabras cifradas en una lista.

```
from nose.tools import assert_equals
from reducer import reducerEV1

def test_reducer1():
    assert_equals(reducerEV1("No Cifrado \n Cifrado123 \n Casa"), 1)

def test_reducer2():
    assert_equals(reducerEV1("NoGFifrado \n Raton \n Teclado \n Pru3b4"), 2)
```

Listado 5.3 Pruebas unitarias de la función reduce (evaluación 1)

Como pasaba en el anterior caso, las pruebas fallarán en un primer momento, pero el código irá emergiendo de forma que al final se obtenga la correcta implementación de la función reduce, la cual, se puede ver en el listado 5.4.

<pre>from operator import itemgetter import sys def reducerEV1(lines): f = open("../EV1_Cifrado/palabras.txt") dic = f.read() vale = True validas = 0 noValidas = 0 for line in lines.split("\n"): vale = True line = line.strip() words = line.split(" ") for word in words: word = word.replace("\\", "/") word = word.lower() if dic.find(word) == -1: vale = False if vale == False: validas += 1 else: noValidas += 1 f.close() return validas</pre>	<pre>from operator import itemgetter import sys f = open("../EV1_Cifrado/palabras.txt") dic = f.read() vale = True validas = 0 noValidas = 0 for line in sys.stdin: vale = True line = line.strip() words = line.split(" ") for word in words: word = word.replace("\\", "/") word = word.lower() if dic.find(word) == -1: vale = False if vale == False: validas += 1 else: noValidas += 1 f.close() print "Cifradas: %d\nNo Cifradas: %d" % (validas, noValidas)</pre>
--	---

Listado 5.4 Función reduce de la evaluación 1

Igual que en el caso del mapper, será necesario realizar una serie de cambios para adaptarlo a las necesidades del MapReduce, como se puede ver en el código de al lado. Este proceso de TDD indicado en este apartado, se utilizará en todas las evaluaciones a implementar pero será explicado de forma más resumida en las siguientes iteraciones.

Finalmente, sólo quedaría probarlo de forma conjunta, para ello será necesario primero utilizar Sqoop para obtener los datos de entrada, una vez se tienen los datos de entrada se usa el sistema Hadoop para crear los trabajos MapReduce que realizarán la evaluación. En la figura 5.9, se puede ver un resumen de la forma de ejecutar los pasos y los resultados obtenidos.

Paso 1: Sqoop para obtener los datos de la base de datos

```
julio@julio-VirtualBox:~$ sqoop import --connect jdbc:mysql://localhost/datos_madrid -  
-username root --password root -e 'SELECT descripcion_centro from presupuestos WHERE 1  
AND $CONDITIONS' --target-dir /user/julio/input/EV1/descripcion_centro -m 1 --direct
```

Paso 2: Ejecutar con Hadoop los trabajos MapReduce

```
julio@julio-VirtualBox:~$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-str  
eaming-2.6.0.jar -input /user/julio/input/EV1/descripcion_centro/part-m-00000 -file /hom  
e/julio/Escritorio/EvaluacionesTFG/EV1_Cifrado/mapper.py -mapper 'python mapper.py' -fil  
e /home/julio/Escritorio/EvaluacionesTFG/EV1_Cifrado/reducer.py -reducer 'python reducer  
.py' -output /user/julio/output/EV1/descripcion_centro
```

Paso 3: Copiar los resultados a local

```
julio@julio-VirtualBox:~$ hadoop fs -copyToLocal /user/julio/output/EV1/descripcion_cen  
tro/part-00000 ./resultados/
```

Paso 4: Resultados de evaluación

```
Cifradas: 187  
No Cifradas: 4990
```

Figura 5.9 Funcionamiento evaluación 1: cifrado

5.3.5 Revisión del Sprint

A la finalización del sprint, se ha conseguido el segundo incremento de la herramienta, que consiste en la primera evaluación de la misma, cuyo objetivo era observar la cantidad de registros que se encuentran cifrados dentro de una columna de una base de datos. En este caso, todavía no se ha automatizado de forma que se pueda indicar la columna de la base de datos y se ejecute automáticamente. Este será un asunto a tratar en el sprint dedicado a realizar la herramienta web.

Sobre las estimaciones realizadas sobre este sprint, si bien la historia de usuario 12 se tardó menos de lo esperado, la historia de usuario 2 se tardó más, por tanto, quedaron más o menos compensadas y no será necesario realizar una reestructuración temporal.

5.4 Sprint 3

Durante este sprint se llevará a cabo la implementación de la segunda evaluación de la herramienta, cuyo objetivo es comprobar qué cantidad de usuarios tienen un permiso en concreto. En las tablas 5.3 y 5.4 se puede ver un resumen de los objetivos de este sprint.

Para este sprint, será necesario obtener la tabla de la base de datos en la cual se indican los permisos que tiene cada usuario, para luego compararlo con el permiso que se quiere estudiar y así obtener un porcentaje sobre la cantidad de usuarios totales.

5.4.1 Refinamiento de la Pila del Producto

La Pila del Producto no ha sufrido modificaciones tras su revisión.

5.4.2 Planificación del Sprint

En el caso de este sprint, se dispone de la historia de usuario 3 (“Quiero comprobar qué usuarios tienen un cierto privilegio”). Esta historia de usuario sirve para implementar la segunda evaluación del sistema.

Para esta evaluación, primero será necesario conocer en qué tabla de la base de datos se encuentran los datos de los permisos de los usuarios, para luego generar un trabajo MapReduce que compare los usuarios y sus permisos con el permiso que nos interesa, para así conseguir el resultado final.

Finalmente, la pila de sprint está formada por una única historia de usuario, que es la siguiente:

Historia de Usuario	
Número: 3	
Nombre historia: Comprobar permisos de usuarios	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 27,5 horas	Sprint asignado: 3
Programador responsable: Julio Moreno García-Nieto	
Descripción: Implementar la evaluación 2 del sistema correspondiente a evaluar qué porcentaje de usuarios tienen un permiso en concreto.	
Postcondición: La evaluación 2 queda implementada.	

Tareas:

- Obtener los datos sobre los permisos de los usuarios de la base de datos.
- Pruebas unitarias para el mapper.
- Implementar el mapper.
- Pruebas unitarias para el reducer.
- Implementar el reducer.
- Integración de mapper y reducer.

Tabla 5.10 Historia de usuario "Comprobar permisos de usuarios"

5.4.3 Desarrollo de historia de usuario: *Comprobar permisos de usuarios*

Tal como se dijo en la planificación del sprint, lo primero será localizar la tabla en la cual se guardan los datos sobre los permisos de los usuarios. Dicha tabla se encuentra situada en INFORMATION_SCHEMA, que es el lugar donde se guardan los metadatos de la base de datos.

Los metadatos [73] son datos sobre los datos, es decir, información relativa a los datos como el nombre de una base de datos o una tabla, el tipo de datos de una columna o los privilegios de acceso. Esta información también es conocida como diccionario de datos.

En resumen, INFORMATION_SCHEMA es la información de la base de datos, es donde se almacena la información referente al resto de bases de datos pertenecientes al servidor de MySQL. La mayoría de tablas que contiene esta base de datos son de sólo lectura, ya que, en realidad se trata de vistas de una tabla. Esto se hace para proteger estos metadatos. Dentro de estos metadatos, los que interesan para la realización de esta evaluación son aquellos relacionados con los permisos de los usuarios, los cuales, se encuentran almacenados en la tabla USER_PRIVILEGES. En la figura 5.10 se puede ver el formato que tiene esta tabla.

The screenshot shows a MySQL Workbench interface with three tabs open. The central tab displays the results of a SELECT query on the USER_PRIVILEGES table. The table has four columns: User, Privilege, Type, and授予权限 (Grant). The data shows multiple entries for users 'julio' and 'julio2' with various privileges like USAGE, SELECT, INSERT, UPDATE, DELETE, CREATE, and DROP.

User	Privilege	Type	授予权限 (Grant)
'julio'@'localhost'	def	USAGE	NO
'julio2'@'localhost'	def	SELECT	NO
'julio2'@'localhost'	def	INSERT	NO
'julio2'@'localhost'	def	UPDATE	NO
'julio2'@'localhost'	def	DELETE	NO
'julio2'@'localhost'	def	CREATE	NO
'julio2'@'localhost'	def	DROP	NO

Figura 5.10 SELECT sobre la tabla USER_PRIVILEGES

Una vez se tienen los datos de privilegios, se van a implementar las funciones map y reduce. Como en el anterior sprint, se utilizará la metodología TDD, por tanto, lo primero será crear una serie de pruebas unitarias, las cuales, se pueden ver en el listado 5.5.

```
from nose.tools import assert_equals
from mapper import mapperEV2

def test_mapper():
    assert_equals_(mapperEV2("'julio'@'localhost', def, DELETE"), "'julio'@'localhost'\tDELETE")
```

Listado 5.5 Pruebas unitarias de la función map (evaluación 2)

Tras una serie refactorizaciones de código finalmente se obtiene la función map que se puede ver en el listado 5.6. Cabe destacar que en este caso la obtención de la entrada para el mapper realizada con la herramienta Sqoop, se ha realizado sobre toda la tabla, por tanto la función map debe formatear la entrada quedándose sólo con lo necesario.

```
#!/usr/bin/env python
|
import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split(',')

    print '%s\t%s' % (words[0], words[2])
```

Listado 5.6 Función map de la evaluación 2

Una vez se tiene la función map, creamos las pruebas necesarias para crear la función reduce (listado 5.7). Tras varias refactorizaciones se obtiene la función reduce, la cual, obtiene como entrada una lista ordenada de usuarios con sus respectivos permisos, para luego comprobar qué usuarios tienen el permiso buscado, obteniendo como salida un número de usuarios con permiso y sin permiso. En el listado 5.8 se puede ver el código de la función reduce resultante.

```
from nose.tools import assert_equals
from reducer import reducerEV2

def test_reducer1():
    assert_equals(reducerEV2("'julio'@'localhost'\tDELETE\n'julio2'@'localhost'\tDELETE\n'julio2'@'localhost'\tSELECT", "DELETE"), 2)

def test_reducer2():
    assert_equals(reducerEV2("'julio'@'localhost'\tINSERT\n'julio2'@'localhost'\tDELETE\n'julio2'@'localhost'\tSELECT", "INSERT"), 1)
```

Listado 5.7 Pruebas unitarias de la función reduce (evaluación 2)

```

#!/usr/bin/env python

from operator import itemgetter
import sys

permisoBuscado = sys.argv[1]
contTotal = -1
contParcial = 0
user = None
actualUser = None

for line in sys.stdin:
    line = line.strip()
    user, permiso = line.split("\t")

    if actualUser != user:
        actualUser = user
        contTotal += 1

    if permiso == permisoBuscado:
        contParcial += 1

print "Con permiso: %d\nSin permiso: %d" % (contParcial, (contTotal-contParcial))

```

Listado 5.8 Función reduce de la evaluación 2

Finalmente, queda comprobar que los trabajos funcionan al trasladarlos al sistema Hadoop. En la figura 5.11 se pueden ver los pasos necesarios para realizarlo y el resultado final.

Paso 1: Sqoop para obtener los datos de la base de datos

```
julio@julio-VirtualBox:~$ sqoop import --connect jdbc:mysql://localhost/information_schema --username root --password root --table user_privileges --target-dir /user/julio/input/EV2/DELETE -m 1
```

Paso 2: Ejecutar con Hadoop los trabajos MapReduce

```
julio@julio-VirtualBox:~$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.0.jar -input /user/julio/input/EV2/DELETE/part-m-00000 -file /home/julio/Escritorio/EvaluacionesTFG/EV2_PermisosUser/mapper.py -mapper 'python mapper.py' -file /home/julio/Escritorio/EvaluacionesTFG/EV2_PermisosUser/reducer.py -reducer 'python reducer.py' -output /user/julio/output/EV2/DELETE
```

Paso 3: Copiar los resultados a local

```
julio@julio-VirtualBox:~/Escritorio/Interfaz$ hadoop fs -copyToLocal /user/julio/output/EV2/DELETE/part-00000 ./resultados/
```

Paso 4: Resultados de evaluación

```
Con permiso: 8
Sin permiso: 0
```

Figura 5.11 Funcionamiento de la evaluación 2: permisos

5.4.4 Revisión del Sprint

A la finalización del sprint, se ha conseguido el tercer incremento de la herramienta, obteniendo la segunda evaluación, cuyo objetivo era observar la cantidad de usuarios

que cumplen un permiso concreto. Para ello, ha sido necesario estudiar los metadatos de una base de datos, así como su estructura.

Al igual que en la anterior evaluación, no se encuentra completamente automatizado, ya que, aunque no es necesario indicar la tabla necesaria (será siempre USER_PRIVILEGES) sí que será necesario poner cuál es el permiso a evaluar. El asunto de la automatización será tratado en el sprint dedicado a realizar la herramienta web.

Sobre la estimación realizada sobre este sprint, se ha cumplido casi en su totalidad.

5.5 Sprint 4

Durante este sprint se llevará a cabo la implementación de la tercera evaluación de la herramienta, cuyo objetivo es evaluar si un usuario en concreto cumple con la cantidad de accesos por hora que se le suponen. En las tablas 5.3 y 5.4 se puede ver un resumen de los objetivos de este sprint.

Para este sprint, será necesario estudiar la tabla de la base de datos, en la que, se guardan el número de accesos que puede realizar, para luego intentar lanzar una prueba contra la base de datos, en la cual, se intente hacer el número de accesos supuestos. En caso de que no sea posible, saltará un error, el cual, se recogerá y servirá para confirmar que no puede realizar tantos accesos por hora.

5.5.1 Refinamiento de la Pila del Producto

La Pila del Producto no ha sufrido modificaciones tras su revisión.

5.5.2 Planificación del Sprint

En el caso de este sprint, se tiene la historia de usuario 4 (“Quiero comprobar que un usuario sólo puede realizar un número máximo de accesos”). Esta historia de usuario sirve para implementar la tercera evaluación del sistema.

Para esta evaluación, primero será necesario estudiar en qué tabla de la base de datos se encuentran la información correspondiente con el máximo número de accesos por hora al sistema, para luego implementar un algoritmo que lance n conexiones a la base de datos con dicho usuario, si hay error implicará que no se pueden realizar tantas conexiones a la base de datos.

Durante este sprint se cambia la forma de implementar la evaluación, ya que, en este caso no se considera necesario utilizar Big Data para lograr el objetivo porque para esta evaluación en concreto, no se va a tratar con grandes cantidades de datos. Con ello, se quiere demostrar que si bien el Big Data es una tecnología muy potente y útil, no es válida para todas las situaciones y será necesario un estudio previo para determinar cuando usarlo.

Finalmente, la pila de sprint está formada por una única historia de usuario, que es la siguiente:

Historia de Usuario	
Número: 3	
Nombre historia: Comprobar máximo de conexiones de usuario	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 21 horas	Sprint asignado: 4
Programador responsable: Julio Moreno García-Nieto	
Descripción: Implementar la evaluación 3 del sistema cuyo objetivo es evaluar si es posible realizar un número de conexiones n a la base de datos.	
Postcondición: La evaluación 3 queda implementada.	
Tareas: <ul style="list-style-type: none"> • Estudiar en qué lugar de la base de datos se almacena la información referida al número máximo de accesos por hora. • Pruebas unitarias para el algoritmo. • Implementar el algoritmo. 	

Tabla 5.11 Historia de usuario "Comprobar máximo de conexiones de usuario"

5.5.3 Desarrollo de historia de usuario: *Comprobar número máximo de accesos de un usuario*

Como se dijo en la planificación del sprint lo primero a realizar para completar esta historia de usuario será buscar dónde se encuentran almacenados los datos correspondientes al máximo número de conexiones permitidas por usuario. Esto se busca no para obtener los datos, sino como forma de comprobar que la evaluación se está realizando de forma correcta.

Esta información no se encuentra situada en la base de datos INFORMATION_SCHEMA como en el caso de los permisos, sino que se encuentra en la base de datos MYSQL y dentro de ella en la tabla user.

Para realizar las pruebas necesarias para la evaluación es necesario tener claro cuántas conexiones puede realizar un usuario en concreto, por ello, se van a crear una serie de

usuario a los cuales se les van a modificar el número de conexiones que pueden realizar. En la figura 5.12 se puede observar cuál es la query necesaria para realizar los cambios en el máximo de conexiones y como quedaría el número máximo de conexiones en la tabla user, el 0 indica que no existen límites para dicho usuario.

```

Archivo Edición Pestañas Ayuda
julio@julio-V... × julio@julio-V... × julio@julio-V... ×
mysql> grant all on *.* to 'julio'@'localhost' with MAX_CONNECTIONS_PER_HOUR 5;
+-----+
| User      | max_connections |
+-----+
| root      |          0        |
| debian-sys-maint |          0        |
| julio     |          5        |
| julio2    |          4        |
| julio3    |         10       |
+-----+
8 rows in set (0.00 sec)

```

Figura 5.12 Máximo de conexiones por usuario

Una vez se tienen el número máximo de conexiones, se procede a diseñar las pruebas unitarias propias del TDD. Dichas pruebas utilizarán los datos obtenidos de la consulta a la base anterior y se pueden ver en el listado 5.9, como se comentó en la planificación en este caso no se va a utilizar técnicas de Big Data para implementar la evaluación, por tanto, sólo será necesario implementar un algoritmo que lance conexiones a la base de datos. El código resultante se puede ver en el listado 5.10.

```

from nose.tools import assert_equals
from maxAccesos import evaluacion3

def test_evaluacion3():
    assert_equals(evaluacion3("julio", "julio", 4, "localhost", "datos_madrid"), "Evaluacion correcta.")

def test_evaluacion3_2():
    assert_equals(evaluacion3("julio2", "julio2", 5, "localhost", "datos_madrid"), "Error."]

```

Listado 5.9 Pruebas unitarias de la evaluación 3

```

import mysql.connector
import sys
import time

user = sys.argv[1]
pwd = sys.argv[2]
max_connections = int(sys.argv[3])
host = sys.argv[4]
database = sys.argv[5]
|
f=open('/home/julio/Escritorio/Interfaz/resultados/EV3'+user, 'w')

for i in range(1, max_connections):
    try:
        con = mysql.connector.connect(user=user, password=pwd, host=host, database=database)
    except mysql.connector.Error, e:
        f.write("Error.")
        time.sleep(5)
        sys.exit(0)

f.write("Evaluacion correcta.")
|

```

Listado 5.10 Código correspondiente a la evaluación 3

Finalmente como salida del algoritmo se obtendrá un archivo resultado, en el cual, se indicará si la evaluación ha sido correcta o no.

5.5.4 Revisión del Sprint

A la finalización del sprint, se ha conseguido el cuarto incremento de la herramienta, obteniendo como resultado la tercera evaluación, cuyo objetivo era comprobar si se cumplen el número máximo de conexiones que se le suponen a un usuario. Para ello, ha sido necesario estudiar la estructura básica de una base de datos hasta encontrar dónde se encontraban dichos datos. Además, para esta evaluación a diferencia de las anteriores, no se ha considerado necesario hacer uso de Big Data al no tratarse de una gran cantidad de datos.

La automatización al igual que en las anteriores evaluaciones será implementada en el sprint dedicado a realizar la herramienta web, ya que, en el caso de esta evaluación será necesario introducir los datos de acceso del usuario que se quiere comprobar y el número de conexiones que se le suponen.

Sobre la estimación realizada sobre este sprint es necesario comentar que se ha quedado un poco corta y, por tanto, ha requerido más tiempo para su realización pero estas horas extras se han realizado en el mismo día, por tanto, no la planificación temporal no se ve afectada.

5.6 Sprint 5

Durante este sprint se llevará a cabo la implementación de la cuarta evaluación de la herramienta, cuyo objetivo será comprobar que a los usuarios que ya no pertenecen al sistema se les han retirado los permisos. En las tablas 5.3 y 5.4 se puede ver un resumen de los objetivos de este sprint.

Para implementar esta evaluación, será necesario que el auditor introduzca un archivo al sistema con los datos de los usuarios que ya no se encuentran en el sistema, para posteriormente realizar la evaluación.

5.6.1 Refinamiento de la Pila del Producto

Se ha decidido hacer un pequeño cambio en la Pila del Producto a la hora de realizar este sprint, ya que, se va a modificar el orden en el que se implementan las historias de usuario, así se realizará primero la historia de usuario correspondiente a introducir un archivo con los usuarios a evaluar (historia de usuario 13) para posteriormente hacer la de comprobar que los usuarios que ya no pertenecen al sistema no disponen de permisos (historia de usuario 5).

5.6.2 Planificación del Sprint

Para este sprint, se dispone de dos historias de usuario: la historia de usuario 13 (“Quiero introducir un archivo con los usuarios eliminados del sistema.”) y la historia de usuario 5 (“Quiero comprobar que los usuarios que ya no pertenecen al sistema ya no disponen de permisos.”).

La historia de usuario 13 consiste en disponer de un archivo con los usuarios ya eliminados, este archivo quedará almacenado en el servidor. También será necesario tener otro archivo con los permisos de los usuarios actuales del sistema para poder compararlas. En resumen, con esta historia de usuario se busca obtener la entrada para realizar la evaluación.

Por otro lado, la historia de usuario 5 tiene como objetivo implementar la cuarta evaluación del sistema, la cual, consiste en evaluar si los usuarios del archivo de entrada siguen teniendo permisos en el sistema. Para ello, será necesario realizar una serie de pruebas unitarias para luego crear los trabajos MapReduce que realicen la evaluación.

Finalmente, la pila de sprint está formada dos historias de usuario, que son las siguientes:

Historia de Usuario	
Número: 13	
Nombre historia: Introducir archivo con usuarios eliminados	
Valor de negocio: Bajo	Riesgo en desarrollo: Bajo
Esfuerzo: 3 horas	Sprint asignado: 5
Programador responsable: Julio Moreno García-Nieto	
Descripción: Añadir un archivo con los usuarios ya eliminados del sistema, también se obtendrán los datos de los permisos de los usuarios actuales; ambos configurarán la entrada de la evaluación.	
Postcondición: El archivo de usuarios eliminados queda añadido al sistema.	
Tareas:	
<ul style="list-style-type: none"> • Crear archivo con los usuarios eliminados del sistema. • Obtener de la base de datos los permisos de los usuarios actuales. 	

Tabla 5.12 Historia de usuario "Introducir archivo con usuarios eliminados"

Historia de Usuario	
Número: 5	
Nombre historia: Comprobar qué usuarios eliminados no disponen de permisos	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 27,5 horas	Sprint asignado: 5
Programador responsable: Julio Moreno García-Nieto	
Descripción: Implementar la evaluación 4 del sistema correspondiente a evaluar qué porcentaje de usuarios eliminados sigue teniendo permisos.	
Postcondición: La evaluación 4 queda implementada.	
Tareas:	
<ul style="list-style-type: none"> • Pruebas unitarias para el mapper. • Implementar el mapper. • Pruebas unitarias para el reducer. • Implementar el reducer. • Integración de mapper y reducer. 	

Tabla 5.13 Historia de usuario "Comprobar qué usuarios eliminados no disponen de permisos"

5.6.3 Desarrollo de historia de usuario: *Introducir archivo con los usuarios a comprobar*

Como se comentó en la planificación del sprint, durante esta historia de usuario se obtendrán los datos de entrada para la evaluación 4. Para ello, lo primero será crear un archivo con los usuarios eliminados del sistema, este archivo será subido por el auditor a la herramienta web.

Dicho archivo consistirá en una lista con los nombres de los usuarios a evaluar, en este caso no es necesario añadir su contraseña. Para ello, se seguirá el formato con el que se guardan los usuarios en la base de datos. En la figura 5.13 se puede ver un pequeño ejemplo de archivo con los usuarios a evaluar.

```
'root'@'linux'  
'root'@'localhost'  
'julio'@'localhost'  
'pepito'@'localhost'  
'julio3'@'localhost'  
'invento'@'localhost'
```

Figura 5.13 Archivo con los usuarios eliminados del sistema

Una vez tenemos creado dicho archivo, será necesario obtener los datos de los permisos de los usuarios, para lo cual, se utilizará el mismo comando usado en el sprint 3. Con esto, conseguimos los datos de entrada necesarios para realizar la evaluación.

5.6.4 Desarrollo de historia de usuario: *Comprobar si a los usuarios eliminados se les han revocado los permisos*

Esta historia de usuario básicamente consiste en implementar las funciones map y reduce. Como en los anteriores sprints, se utilizará la metodología TDD, por tanto, lo primero será crear las pruebas unitarias. Las pruebas de la función map se pueden observar en el listado 5.11.

```
from nose.tools import assert_equals  
from mapper import mapperEV4  
  
def test_mapper():  
    assert_equals(mapperEV4("julio'@'localhost", def, DELETE), "julio'@'localhost'\tDELETE")
```

Listado 5.11 Pruebas unitarias de la función map (evaluación 4)

```

import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split(',')
    print '%s\t%s' % (words[0], words[2])

```

Listado 5.12 Función map de la evaluación 4

Tras una serie refactorizaciones de código finalmente se obtiene la función map que se puede ver en el listado 5.12. Al igual que en el sprint 3, no es necesario formatear la entrada que le llega al mapper, que sólo se encargará de transmitir esa información.

Una vez se tiene la función map, se crean las pruebas necesarias para crear la función reduce (listado 5.13). Tras varias refactorizaciones se obtiene la función reduce, la cual, obtiene como entrada una lista ordenada de usuarios con sus respectivos permisos, también tiene como entrada el archivo con los usuarios a evaluar. Por tanto, el objetivo de la función reduce será comprobar si cada usuario de la base de datos se encuentra en el archivo con los usuarios eliminados. Como salida se obtendrá el número de usuarios supuestamente eliminados, que siguen teniendo permisos. En el listado 5.14 se puede ver el código de la función reduce resultante.

```

from nose.tools import assert_equals
from reducer import reducerEV4

def test_reducer1():
    assert_equals(reducerEV4("julio@localhost\tDELETE\njulio2@localhost\tDELETE\njulio2@localhost\tSELECT"), 1)
def test_reducer2():
    assert_equals(reducerEV4("julio@localhost\tINSERT\njulio2@localhost\tDELETE\njulio3@localhost\tSELECT"), 2)

```

Listado 5.13 Pruebas unitarias de la función reduce (evaluación 4)

```

from operator import itemgetter
import sys

f = open(sys.argv[1])
users = f.read()
userActual = None
contador = 0

total = len(users.split('\n'))

for line in sys.stdin:
    line = line.strip()
    user, permiso = line.split("\t")

    if users.find(user) != -1:
        if (user != userActual):
            userActual = user
            contador = contador + 1

f.close()

print "Usuarios con permisos no revocados: %d\nUsuarios sin permisos: %d" % (contador, (total-contador))

```

Listado 5.14 Función reduce de la evaluación 4

Finalmente, en la figura 5.14 se puede observar los pasos necesarios para ejecutar la evaluación usando el sistema Hadoop.

Paso 1: Sqoop para obtener los datos de la base de datos

```
julio@julio-VirtualBox:~$ sqoop import --connect jdbc:mysql://localhost/information_schema --username root --password root --table user_privileges --target-dir /user/julio/input/EV4/users -m 1
```

Paso 2: Ejecutar con Hadoop los trabajos MapReduce

```
julio@julio-VirtualBox:~$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.0.jar -input /user/julio/input/EV4/users/part-m-00000 -file /home/julio/Escritorio/EvaluacionesTFG/EV4_UsersEliminados/mapper.py -mapper 'python mapper.py' -file /home/julio/Escritorio/EvaluacionesTFG/EV4_UsersEliminados/reducer.py -reducer 'python reducer.py' -output /user/julio/output/EV4/users
```

Paso 3: Copiar los resultados a local

```
julio@julio-VirtualBox:~$ hadoop fs -copyToLocal /user/julio/output/EV4/users/part-00000 ./resultados/
```

Paso 4: Resultados de evaluación

```
Usuarios con permisos no revocados: 3  
Usuarios sin permisos: 3
```

Figura 5.14 Funcionamiento de la evaluación 4: usuarios eliminados

5.6.5 Revisión del Sprint

A la finalización del sprint, se ha conseguido el quinto incremento de la herramienta, obteniendo como resultado la cuarta evaluación, cuyo objetivo era comprobar si los usuarios que ya no pertenecen al sistema siguen teniendo permisos en el mismo. Para ello, ha sido necesario añadir un archivo con los usuarios eliminados del sistema y crear un trabajo MapReduce. Para esta evaluación se ha reutilizado información obtenida en el sprint 3, referida a como obtener los datos de los permisos de los usuarios.

La automatización al igual que en las anteriores evaluaciones será implementada en el sprint dedicado a realizar la herramienta web, ya que, en el caso de esta evaluación será necesario introducir el archivo con los usuarios eliminados del sistema.

Sobre la estimación temporal realizada para este sprint, es necesario recalcar que fue demasiado pesimista, probablemente porque no se tuvo en cuenta la posibilidad de reutilizar información de otros sprints, y por tanto, se ha realizado en menor tiempo del esperado.

5.7 Sprint 6

Durante este sprint se llevará a cabo la implementación de la quinta evaluación de la herramienta, cuyo objetivo será comprobar que los usuarios sólo acceden al sistema de bases de datos en su horario de trabajo. En las tablas 5.3 y 5.4 se puede ver un resumen de los objetivos de este sprint.

Para implementar esta evaluación, será necesario introducir el archivo de log de la base de datos, además habrá que introducir al sistema un archivo con los usuarios y sus horarios de acceso al sistema.

5.7.1 Refinamiento de la Pila del Producto

Al igual que en los sprints anteriores, en los cuales, había una historia de usuario correspondiente a introducir un archivo al sistema, en este caso, también ha sido necesario modificar el orden de realización de las historias de usuario realizando primero la introducción de un archivo de log y la correspondiente a crear un archivo con los horarios de los usuarios, para finalmente realizar las funciones map y reduce.

5.7.2 Planificación del Sprint

Para este sprint, se dispone de tres historias de usuario: la historia de usuario 9 (“Quiero poder introducir un archivo con el log de la base de datos en el sistema.”), la historia de usuario 14 (“Quiero introducir un archivo con los usuarios a evaluar y su horario de trabajo.”) y la historia de usuario 6 (“Quiero evaluar que una serie de usuarios acceden sólo durante las horas de trabajo.”).

La historia de usuario 9 consiste en disponer del archivo de log del sistema, el cual, servirá como entrada de la evaluación. Por otro lado, la historia de usuario 14 tiene como objetivo introducir un archivo con los horarios de acceso de los usuarios a evaluar, para ello, será necesario definir una estructura para el archivo que la función reduce sea capaz de interpretar.

Una vez tenemos las entradas necesarias para la realización de la evaluación, se implementará la historia de usuario 14, cuyo objetivo principal es el de crear los trabajos MapReduce necesarios, para ello, se crearán unas pruebas unitarias.

Finalmente, la pila de sprint está formada tres historias de usuario, que son las siguientes:

Historia de Usuario	
Número: 9	
Nombre historia: Introducir archivo de log	
Valor de negocio: Medio	Riesgo en desarrollo: Medio
Esfuerzo: 5 horas	Sprint asignado: 6
Programador responsable: Julio Moreno García-Nieto	
Descripción: Añadir un archivo con el log del sistema. También será necesario realizar un estudio de su estructura para saber qué datos son interesantes para la realización de la evaluación.	
Postcondición: El archivo de log queda añadido al sistema	
Tareas: <ul style="list-style-type: none"> • Localizar archivo de log del sistema. • Estudiar su estructura. 	

Tabla 5.14 Historia de usuario "Introducir archivo de log"

Historia de Usuario	
Número: 14	
Nombre historia: Introducir archivo con horario de los usuarios	
Valor de negocio: Bajo	Riesgo en desarrollo: Bajo
Esfuerzo: 5 horas	Sprint asignado: 6
Programador responsable: Julio Moreno García-Nieto	
Descripción: Añadir un archivo con el horario de acceso al sistema que tienen los usuarios, será necesario definir una estructura para dicho archivo.	
Postcondición: El archivo con el horario de acceso de los usuarios queda añadido al sistema.	
Tareas: <ul style="list-style-type: none"> • Definir estructura para el archivo. • Crear archivo con horario de usuarios. 	

Tabla 5.15 Historia de usuario "Introducir archivo con horario de los usuarios"

Historia de Usuario	
Número: 6	
Nombre historia: Evaluar que los usuarios sólo acceden en sus horas de trabajo	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 27,5 horas	Sprint asignado: 6
Programador responsable: Julio Moreno García-Nieto	
Descripción: Implementar la evaluación 5 del sistema correspondiente a evaluar si los usuarios acceden al sistema sólo durante sus horas de trabajo.	

Postcondición:

La evaluación 5 queda implementada.

Tareas:

- Pruebas unitarias para el mapper.
- Implementar el mapper.
- Pruebas unitarias para el reducer.
- Implementar el reducer.
- Integración de mapper y reducer.

Tabla 5.16 Historia de usuario "Evaluar que los usuarios sólo acceden en sus horas de trabajo"

5.7.3 Desarrollo de historia de usuario: *Introducir archivo log*

Para la implementación de la evaluación 5 se produce un cambio en los datos que se toman para su estudio, ya que en este caso no se evaluará unos datos procedentes de la base de datos en sí, sino que se utilizará el fichero de log del sistema de bases de datos.

Lo primero será localizar el archivo log, el cual, tendrá que haber sido habilitado en el sistema para que se encuentre disponible. Una vez localizado, se estudiará su estructura para observar en qué líneas del archivo se indica cuando se ha realizado un acceso, esos serán los datos que se necesitarán para comprobar la hora y si esta se encuentra dentro del horario permitido. En la figura 5.15 se puede observar un ejemplo archivo de log y señalado un acceso.

```
40 Query SELECT `GRANTEE`, `TABLE_CATALOG`, `PRIVILEGE_TYPE`, `IS_GRANTABLE` FROM `user_privileges
AND ( 1=1 )
40 Query SET net_write_timeout=60
40 Query commit
40 Query rollback
40 Quit |
150703 20:28:32 41 Connect root@localhost on
41 Query select @version_comment limit 1
150703 20:28:34 41 Quit
150703 20:28:36 42 Connect rooy@localhost on
42 Connect Access denied for user 'rooy'@'localhost' (using password: YES)
150703 20:28:38 43 Connect julio@localhost on
43 Query select @version_comment limit 1
150703 20:28:40 43 Quit
150703 20:28:41 44 Connect julio@localhost on
44 Query select @version_comment limit 1
150703 20:28:43 44 Quit
150703 20:28:45 45 Connect julio@localhost on
45 Connect Access denied for user 'julio'@'localhost' (using password: YES)
150703 20:28:47 46 Connect julio@localhost on
46 Connect Access denied for user 'julio'@'localhost' (using password: YES)
150703 20:28:48 47 Connect julio2@localhost on
47 Connect Access denied for user 'julio2'@'localhost' (using password: YES)
150703 20:28:49 48 Connect julio3@localhost on
```

Figura 5.15 Ejemplo de archivo de log

5.7.4 Desarrollo de historia de usuario: *Introducir archivo con horarios de los usuarios*

Como se comentó en la planificación del sprint, la entrada necesaria para la evaluación 5 se compone de dos partes: el archivo de log y el archivo con los horarios de los usuarios. Dicho archivo servirá para que la función reducer compruebe que cada uno de los accesos realizados a la base de datos se han producido o no en su horario permitido.

Para ello, será fundamental la definición de una estructura que sea comprensible para la función reducer. Esta estructura se ha decidido que tenga la siguiente forma:

```
<nombre_de_usuario> in: <hora_entrada> // out: <hora_salida>
```

Este archivo tendrá que ser introducido al sistema por el auditor, al cual, se le darán las instrucciones necesarias para la creación de este fichero en la herramienta web.

5.7.5 Desarrollo de historia de historia de usuario: *Evaluuar accesos de usuario en sus horas de trabajo*

Para el desarrollo de esta historia de usuario se implementarán las funciones map y reduce. Como en los anteriores sprints, se utilizará la metodología TDD siendo necesario la creación de las pruebas unitarias. Las pruebas de la función map se pueden observar en el listado 5.15.

```
from nose.tools import assert_equals
from mapper import mapperEV6

def test_mapper():
    assert_equals(mapperEV6("150703 20:28:45      45 Connect  julio@localhost on
\n 45 Connect  Access denied for user 'julio'@'localhost' (using password: YES)"), "julio@localhost\t20:28:45")

def test_mapper2():
    assert_equals(mapperEV6("150703 20:28:41      44 Connect  julio@localhost on
\n 44 Query  select @@version comment limit 1"), "julio@localhost\t20:28:41")
```

Listado 5.15 Pruebas unitarias de la función map (evaluación 5)

Tras refactorizar el código se obtiene la función map que se puede ver en el listado 5.16. En este caso, si será necesario realizar un formateo de la entrada del archivo de log, ya que, sólo interesa quedarse con las líneas referentes a la realización de un acceso al sistema, por ello, se puede ver que se ha hecho una comprobación de cada línea.

En caso de que fuese un acceso al sistema, sólo es útil la información referida al usuario que accedió y su hora de conexión, por tanto, ahí se establece un nuevo formateo de la entrada.

```

import sys

user = ''
hour = ''

for line in sys.stdin:
    line = line.strip()
    words = line.split('\t')

    if (line.find('Connect') != -1):
        if (len(words) > 2) :
            hour = words[0].split(" ")[1]
            user = words[2].replace('on', '')
            print "%s\t%s" % (user, hour)

```

Listado 5.16 Función map de la evaluación 5

Con la función map ya implementada, se definen las pruebas unitarias para crear la función reduce (listado 5.17). Tras refactorizar el código se obtiene la función reduce, la cual, tiene como entrada una lista ordenada con las horas de los accesos de los usuarios, también se dispone como entrada el archivo con los horarios de acceso de dichos usuarios a evaluar.

```

from nose.tools import assert_equals
from reducer import reducerEV5

def test_reducer():
    assert_equals(reducerEV5("julio@localhost\t20:28:41\njulio@localhost\t14:05:32\njulio2@localhost\t15:25:36"),
    "Numero de accesos dentro de horario: 2 |\nNumero de accesos fuera de horario: 1 |\nNumero de usuarios no encontrados: 0 |")

```

Listado 5.17 Pruebas unitarias de la función reduce (evaluación 5)

Por tanto, el objetivo de la función reduce será comprobar si cada acceso a la base de datos se ha realizado dentro del horario permitido, para ello, se realizará una comprobación en la cual observará si la hora de acceso se encuentra dentro de los límites marcados.

Como salida se obtendrá un número de accesos dentro de horario, otro de accesos que fuera de horario y finalmente, el número de accesos de usuarios no encontrados, es decir, que cuyos horarios no se encontraban dentro del archivo. En el listado 5.18 se puede ver el código de la función reduce resultante.

```

from operator import itemgetter
import sys
from datetime import datetime

user = None
words = None
hour = None
hourIn = False
hourOut = False
notFoundUsers = 0
goodAccess = 0
badAccess = 0

for line in sys.stdin:
    userFound = False
    line = line.strip()
    user, hour = line.split("\t")
    hour = datetime.strptime(hour, "%H:%M:%S")

    with open(sys.argv[1], 'r') as inFile:
        for lineF in inFile:
            if user in lineF:
                userFound = True
                hourIn = lineF.split("in:")[1].split("||")[0].replace(' ', '')
                hourIn = datetime.strptime(hourIn, "%H:%M")
                hourOut = lineF.split("out:")[1].replace(' ', '').replace('\n', '')
                hourOut = datetime.strptime(hourOut, "%H:%M")
                if not(hour >= hourIn and hour <= hourOut):
                    goodAccess+=1
                else:
                    badAccess+=1

    if not userFound :
        notFoundUsers += 1
print ("Numero de accesos dentro de horario: %s |" % goodAccess)
print ("Numero de accesos fuera de horario: %s |" % badAccess)
print ("Numero de accesos de usuarios no encontrados: %s |" % notFoundUsers)

```

Listado 5.18 Función reduce de la evaluación 5

Finalmente, en la figura 5.16 se pueden observar los pasos necesarios para ejecutar la evaluación usando el sistema Hadoop.

Paso 1: Se copia el archivo de log al sistema Hadoop

```
julio@julio-VirtualBox:~$ hadoop fs -copyFromLocal mysql.log /user/julio/input/EV6/mysql.log
```

Paso 2: Ejecutar con Hadoop los trabajos MapReduce

```
julio@julio-VirtualBox:~$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.0.jar -input /user/julio/input/EV6/mysql.log -file /home/julio/Escritorio/EvaluacionesTFG/EV6_HorarioAcceso/mapper.py -mapper 'python mapper.py' -file /home/julio/Escritorio/EvaluacionesTFG/EV6_HorarioAcceso/reducer.py -reducer 'python reducer.py' -output /user/julio/output/EV6/res_log
```

Paso 3: Copiar los resultados a local

```
julio@julio-VirtualBox:~$ hadoop fs -copyToLocal /user/julio/output/EV6/res_log/part-00000 ./resultados/
```

Paso 4: Resultados de evaluación

```
Numero de accesos dentro de horario: 10 |
Numero de accesos fuera de horario: 4 |
Numero de accesos de usuarios no encontrados: 4 |
```

Figura 5.16 Funcionamiento evaluación 5: accesos en horario permitido

5.7.6 Revisión del Sprint

A la finalización de este sprint, se ha conseguido el sexto incremento de la herramienta correspondiente a la implementación de la quinta evaluación de la misma. Su objetivo es comprobar si los accesos a la base de datos se hacen dentro del horario permitido para el usuario, para ello, ha sido necesario la introducción del fichero de log del sistema de bases de datos, así como la definición de un archivo con los horarios de acceso permitidos para cada usuario.

Una vez se tenían los datos de entrada necesarios, se implementaron las funciones MapReduce que cubren la funcionalidad buscada, como en el resto de sprints se usó TDD para llevar a cabo esta tarea.

Al igual que en los demás sprints, el tema de la automatización será tratado en el sprint dedicado a la creación de la herramienta web, ya que, es necesario que el usuario de la herramienta introduzca los archivos de entrada.

Finalmente, sobre la estimación temporal cabe destacar que no fue lo suficientemente acertada, ya que, fue demasiado positiva y ha sido necesario realizar horas extras para evitar que este contratiempo afectase al resto de la planificación.

5.8 Sprint 7

Durante el desarrollo de este sprint se llevará a cabo la implementación de la sexta evaluación del sistema, la cual, será la última realizada durante el desarrollo de este TFG. Dicha evaluación tiene como objetivo evaluar la cantidad de accesos erróneos que se producen en el sistema, además de quien los realiza. En las tablas 5.3 y 5.4 se puede ver un resumen de los objetivos de este sprint.

Para realizar esta evaluación, será necesario que el auditor introduzca un archivo de log en el sistema, igual que pasaba en el anterior sprint.

5.8.1 Refinamiento de la Pila del Producto

La Pila del Producto no ha sufrido modificaciones tras su revisión.

5.8.2 Planificación del Sprint

Para este sprint, se dispone de una única historia de usuario: la historia de usuario 7 (“Quiero evaluar la cantidad de accesos fallidos realizados sobre la base de datos.”).

La historia de usuario 7 consiste en crear la evaluación 6 del sistema, para ello, será necesario introducir el log en el sistema como en el anterior sprint. Una vez tenemos la entrada al sistema, se implementarán los trabajos MapReduce mediante el uso de la metodología de desarrollo TDD siendo, por tanto, necesario la realización de unas pruebas unitarias de las cuales emerja el código.

Finalmente, la pila de sprint está formada una historia de usuario, la cual, es la siguiente:

Historia de Usuario	
Número: 7	
Nombre historia: Evaluar accesos fallidos a la base de datos	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 27,5 horas	Sprint asignado: 7
Programador responsable: Julio Moreno García-Nieto	
Descripción: Realizar la evaluación 6 de sistema, correspondiente con evaluar el número de accesos fallidos realizados por usuario.	
Postcondición: La evaluación 6 queda implementada.	
Tareas: <ul style="list-style-type: none">• Introducir archivo log del sistema.• Pruebas unitarias para el mapper.• Implementar el mapper.• Pruebas unitarias para el reducer.• Implementar el reducer.• Integración de mapper y reducer.	

Tabla 5.17 Historia de usuario "Evaluar accesos fallidos a la base de datos"

5.8.3 Desarrollo de historia de usuario: *Evaluar accesos fallidos a la base de datos*

Tal y como se indicó en la planificación del sprint, lo primero será introducir el archivo de log en el sistema, para ello se utilizará lo aprendido durante el sprint 6, en el cual, también era necesario el uso de un archivo de log.

Una vez tenemos el log, lo primero será realizar las pruebas unitarias propias de la metodología TDD, dichas pruebas unitarias se pueden ver en el listado 5.19.

```

from nose.tools import assert_equals
from mapper import mapperEV5

def test_mapper():
    assert_equals(mapperEV5("150703 20:28:45      45 Connect  julio@localhost on
\n 45 Connect  Access denied for user 'julio'@'localhost' (using password: YES)"), "julio@localhost")

def test_mapper2():
    assert_equals(mapperEV5("150703 20:28:41      44 Connect  julio@localhost on
\n 44 Query  select @version_comment limit 1"), "")

```

Listado 5.19 Pruebas unitarias de la función map (evaluación 6)

Tras varias iteraciones de refactorización de código se obtiene el código de la función map, en esta ocasión, no se necesita formatear la entrada para obtener las líneas correspondientes a realizar una acceso, sino todo lo contrario, para ello se puede ver en el código que ha sido necesario poner unas ciertas condiciones. El código de esta función se puede ver en el listado 5.20.

```

import sys

anterior = False
user = ''
hora = ''

for line in sys.stdin:
    line = line.strip()
    words = line.split('\t')

    if (line.find('Connect') != -1):
        if (anterior == True):
            print "%s" % (user)
            anterior = False
        else:
            anterior = True
            if (len(words) > 2) :
                user = words[2].replace('on', '')
    else:
        anterior = False

```

Listado 5.20 Función map de la evaluación 6

Una vez se tiene implementado la función map, lo siguiente será implementar la función reduce, al igual que en los casos anteriores, hay que definir una serie de pruebas unitarias, las cuales, se pueden ver en el listado 5.11.

```

from nose.tools import assert_equals
from reducer import reducerEV5

def test_reducer1():
    assert_equals(reducerEV5("julio@localhost\njulio@localhost"), "- 2 - julio|")

def test_reducer2():
    assert_equals(reducerEV5("julio@localhost\nroot@localhost"), "- 1 - julio|\n- 1 - root|")

```

Listado 5.21 Pruebas unitarias de la función reduce (evaluación 6)

Tras refactorizar el código se obtiene como resultado la función reduce que se puede

observar en el listado 5.22. En este caso, como llegan los usuarios ordenados, es necesario contar el número de veces que han realizado un acceso erróneo. Como salida de la evaluación se obtiene una lista de usuarios y el número de accesos erróneos que ha realizado.

```
from operator import itemgetter
import sys

current_word = None
current_count = 0
user = None
|
for line in sys.stdin:
    user = line.strip()

    if current_word == user:
        current_count += 1
    else:
        if (current_word != None):
            print ' - %s - %s |' % (current_count, current_word.replace("@localhost", ""))
        current_count = 1
        current_word = user

print ' - %s - %s |' % (current_count, current_word.replace("@localhost", ""))
```

Listado 5.22 Función reduce de la evaluación 6

Finalmente, sólo queda probar la solución propuesta en conjunto y usando el sistema Hadoop, para ello, se seguirán los pasos indicados en la figura 5. Obteniendo como resultado un archivo con los usuarios y el número de accesos erróneos realizados. Cabe destacar, que para la realización de este sprint, se han simulado una serie de accesos erróneos al sistema con distintos usuarios, para ello, se creó un pequeño script que intentaba accesos a la base de datos.

Paso 1: Se copia el archivo de log al sistema Hadoop

```
julio@julio-VirtualBox:~$ hadoop fs -copyFromLocal mysql.log /user/julio/input/EV5/mysql.log
```

Paso 2: Ejecutar con Hadoop los trabajos MapReduce

```
julio@julio-VirtualBox:~$ hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.0.jar
-input /user/julio/input/EV5/mysql.log -file /home/julio/Escritorio/EvaluacionesTFG/EV5_AccesosErroneos/mapper.py -mapper 'python mapper.py' -file /home/julio/Escritorio/EvaluacionesTFG/EV5_AccesosErroneos/reducer.py -reducer 'python reducer.py' -output /user/julio/output/EV5/res_log
```

Paso 3: Copiar los resultados a local

```
julio@julio-VirtualBox:~$ hadoop fs -copyToLocal /user/julio/output/EV5/res_log/part-00000 ./resultados/
```

Paso 4: Resultados de evaluación

```
- 2 - julio2 |
- 2 - julio3 |
- 2 - julio |
- 4 - root |
```

Figura 5.17 Funcionamiento evaluación 6: accesos incorrectos

5.8.4 Revisión del Sprint

Tras la finalización de este sprint, se consigue el séptimo incremento de la herramienta, que consiste en la realización de la sexta evaluación del sistema, la cual, será la última que se realizará durante el TFG. El objetivo de esta evaluación es comprobar la cantidad de accesos incorrectos que se han realizado en el sistema.

Para ello, será necesario que el auditor introduzca un fichero de log al sistema, para que finalmente, el MapReduce desarrollado durante el sprint cree un fichero con los resultados deseados.

Finalmente, la estimación temporal ha sido acertada, y por tanto, se ha podido completar el sprint en el tiempo previsto.

5.9 Sprint 8

Una vez acabada la implementación de las evaluaciones que forman la herramienta, llega el momento de desarrollar la herramienta web que ensambla todas las piezas que la componen. Se trata del sprint más costoso en cuanto a dedicación temporal de toda la herramienta. En las tablas 5.3 y 5.4, se puede ver un resumen de los objetivos de este sprint.

Durante este sprint, será necesario elegir primero qué tecnologías se van a usar para su desarrollo, para luego diseñar un esqueleto de lo que será la aplicación y finalmente conseguir aunar todas las piezas que forman el puzzle que es la herramienta.

5.9.1 Refinamiento de la Pila del Producto

A la hora de refinar la Pila del Producto se ha decidido realizar un cambio en el orden de implementación de la historia de usuario, ya que, se considera más natural la nueva ordenación, en la cual, se realizará primero la historia de usuario 11 (interfaz de la herramienta) para luego implementar la número 10 (que la aplicación sea herramienta web).

5.9.2 Planificación del Sprint

Para este sprint, se dispone de dos historias de usuario: la historia de usuario 11 (“Quiero que la herramienta tenga una interfaz.”) y la historia de usuario 10 (“Quiero que sea una aplicación web.”).

La historia de usuario 11 tiene como objetivo la creación de la interfaz de la herramienta, se ejecutará primero esta historia de usuario con el objetivo de crear un esqueleto al que luego ir añadiendo funcionalidades de una forma más rápida.

De implementar estas funcionalidades se encargará precisamente la historia de usuario 10, la cual, tendrá como objetivo principal la integración de todas las evaluaciones realizadas en los anteriores sprints. Para ello, se hará uso de scripts que ejecuten los comandos necesarios para ello.

Finalmente, la pila de sprint está formada dos historias de usuario que son las siguientes:

Historia de Usuario	
Número: 11	
Nombre historia: Desarrollar interfaz web	
Valor de negocio: Medio	Riesgo en desarrollo: Medio
Esfuerzo: 17 horas	Sprint asignado: 8
Programador responsable: Julio Moreno García-Nieto	
Descripción: Implementar la interfaz que tendrá la herramienta web a forma de esqueleto que será completado con la siguiente historia de usuario.	
Postcondición: Interfaz de la herramienta web realizada.	
Tareas: <ul style="list-style-type: none"> • Diseñar aspecto que tendrá la interfaz. • Elegir tecnología a utilizar para implementar la interfaz. • Implementar interfaz de la aplicación web. 	

Tabla 5.18 Historia de usuario "Desarrollar interfaz web"

Historia de Usuario	
Número: 10	
Nombre historia: Implementar aplicación web	
Valor de negocio: Medio	Riesgo en desarrollo: Medio
Esfuerzo: 34 horas	Sprint asignado: 8
Programador responsable: Julio Moreno García-Nieto	
Descripción: Realizar la implementación de la herramienta web que integre todas las evaluaciones realizadas en los anteriores sprints.	
Postcondición: La aplicación web queda implementada.	

Tareas:

- Creación base de datos que contenga los datos de las evaluaciones.
- Diferenciación de las evaluaciones en función de sus necesidades.
- Creación de scripts que ejecuten las distintas evaluaciones.
- Recolección de los archivos resultado generados por las evaluaciones.

Tabla 5.19 Historia de usuario "Implementar aplicación web"

5.9.3 Desarrollo de historia de usuario: *Desarrollar interfaz web*

Tal y como se dijo en la planificación, el primer paso para realizar esta historia de usuario será la realización de unos prototipos del diseño que tendrá la interfaz, para ello, se hará uso de la herramienta Balsamiq Mockups, explicada en el capítulo 3. Los diseños que se han realizado se pueden ver en las siguientes figuras.

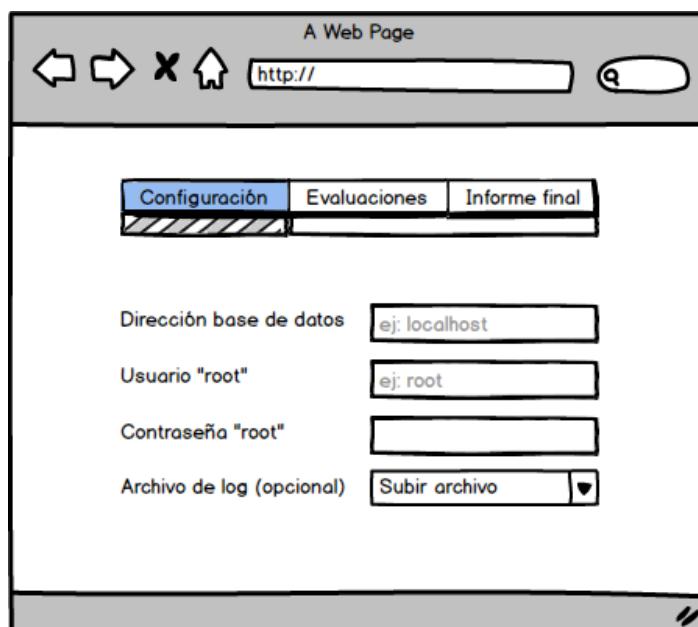


Figura 5.18 Prototipo pestaña de configuración

Como se puede ver, se ha decidido dividir la interfaz en tres distintas pestañas: configuración, evaluaciones e informe final. En esta primera pestaña se puede ver una pantalla para introducir los datos referidos a la base de datos, los cuales serán necesarios para realizar todas las evaluaciones. Por otro lado, también se dispone de un campo para subir el archivo de log que es opcional pero sin el cual no se podrán realizar las evaluaciones 5 y 6.

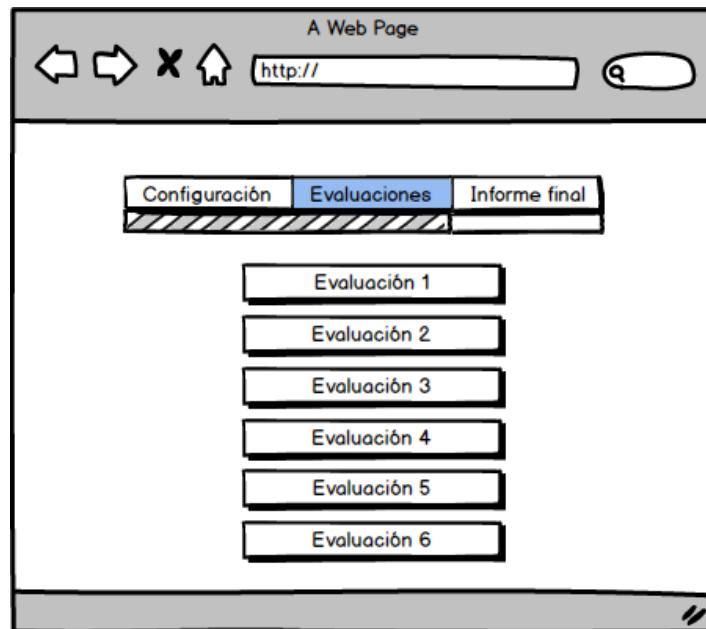


Figura 5.19 Prototipo pestaña de evaluaciones

En esta segunda pestaña se puede observar una lista con todas las evaluaciones del sistema, al pinchar en cada una de ellas llevará al usuario a una nueva pantalla, en la cual, se le pedirán el resto de datos necesarios para realizar dicha evaluación.

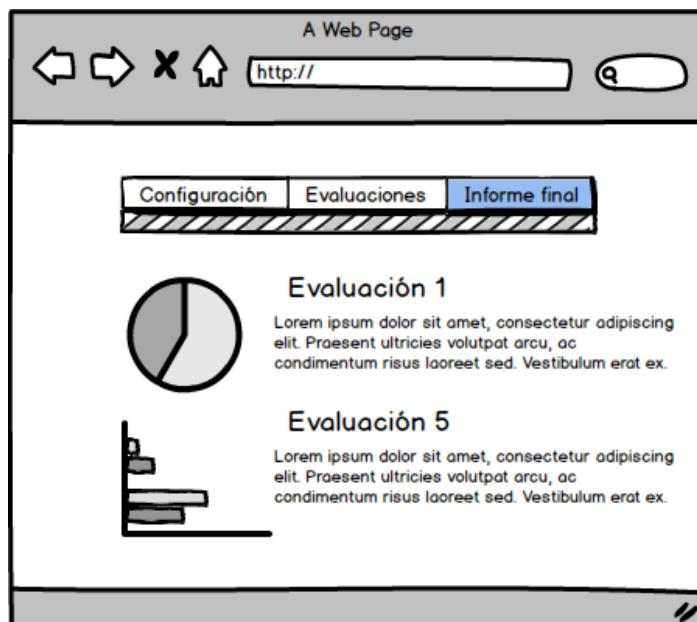


Figura 5.20 Prototipo pestaña de informe final

Finalmente, en esta última pestaña de informe final se presentarán los resultados en forma de gráficos de diferentes tipos en función del tipo de resultado obtenido en la evaluación.

Una vez se tienen los prototipos de la interfaz a desarrollar, la siguiente tarea es elegir con qué tecnología se va a implementar dicha interfaz. Para ello, se ha decidido hacer uso del denominado stack MEAN que integra el backend, el frontend y la base de datos. Para lograrlo, utiliza las tecnologías de Node.js, AngularJS, Express y MongoDB. En la siguiente imagen se puede ver un ejemplo de cómo se relacionan entre sí estas tecnologías.

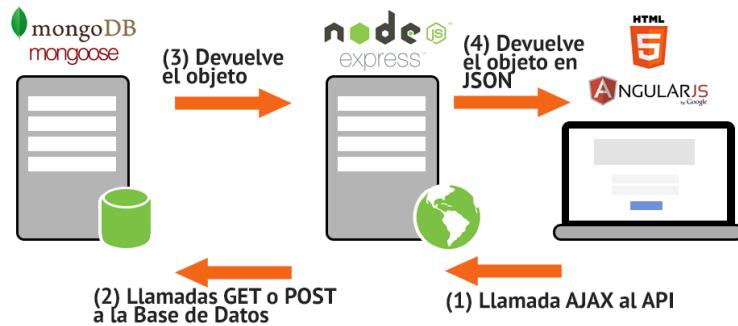


Figura 5.21 Flujo de datos en una aplicación que usa stack MEAN [74]

Con la tecnología ya elegida llega el momento de implementar la interfaz de usuario, lo primero será establecer un árbol de directorios dónde irán almacenados los distintos archivos que formen la interfaz:

- **Evaluaciones TFG:** será la carpeta que contenga los distintos archivos necesarios para realizar las evaluaciones que soporta el sistema.
- **Node_modules:** carpeta donde se almacenan los distintos módulos que se utilizarán para implementar la herramienta web.
- **Public:** carpeta contenedora del archivo HTML, así como de los estilos e iconos de la herramienta.
- **Raiz:** en la raíz de la carpeta se encuentran los archivos correspondientes a la API REST de nodeJS y el archivo con las dependencias necesarias.

Es necesario destacar también antes de empezar a escribir código, que se ha decidido hacer uso de Bootstrap para diseñar la interfaz de una forma más sencilla y obteniendo unos resultados más vistosos (en el capítulo 3 se explica Bootstrap con más detallamiento).

También se ha decidido usar un único archivo HTML, en el cual, se irán mostrando y ocultando las distintas partes del sistema según sea necesario usando directivas propias de AngularJS. Con todo esto y siguiendo los prototipos creados con anterioridad, se ha implementado la siguiente interfaz para la primera pestaña (Configuración).

Evaluación de seguridad en Bases de datos

Configuración inicial Evaluaciones Informe Final

Dirección base de datos	<input type="text" value="ej. localhost"/>
Usuario "root"	<input type="text" value="ej. root"/>
Contraseña "root"	<input type="password"/>
Archivo de log (opcional)	<input type="button" value="Seleccionar archivo"/> Ningún archivo seleccionado <small>Sin subir el archivo de log no podrá realizar algunas evaluaciones</small>

Siguiente

Figura 5.22 Interfaz final pestaña de configuración

Como se puede observar, se ha mantenido el diseño hecho anteriormente, añadiendo simplemente un botón de “Siguiente” para poder avanzar al siguiente paso, también se ha puesto un aviso de que el archivo de log es opcional, pero sino se sube no se podrán realizar algunas evaluaciones.

Evaluación de seguridad en Bases de datos

Configuración inicial **Evaluaciones** Informe Final

01 - Cifrado

02 - Permisos usuarios

03 - Accesos máximos

04 - Usuarios eliminados

05 - Accesos erróneos

06 - Acceso en horario permitido

Volver **Siguiente**

Figura 5.23 Interfaz pestaña de evaluaciones

En la pestaña de evaluaciones, también se ha realizado el cambio de los botones de “Siguiente” y “Volver” respecto al diseño original para facilitar la navegación en la herramienta. Las evaluaciones de momento sólo se trata de botones que no llaman a

ninguna función, esto se realizará en la siguiente historia de usuario.



Figura 5.24 Interfaz pestaña de informe final

Finalmente en el caso de la pestaña de informe final, sólo se ha implementado la pestaña, ya que, en esta pestaña se encontrarán los resultados obtenidos de la realización de las evaluaciones, lo cual, se realizará en el siguiente sprint.

5.9.4 Desarrollo de historia de usuario: *Implementar aplicación web*

Con el esqueleto de la interfaz ya realizado llega el momento de integrar todas las evaluaciones en el sistema y de, por tanto, dejar la herramienta prácticamente acabada, a falta de presentar los resultados de una forma más visual.

El primer paso para ello será crear una base de datos de tipo MongoDB y alojada en MongoLab (estas tecnologías se explican con más detenimiento en el capítulo 3). Esta base de datos servirá para alojar la información necesaria de las evaluaciones, como su nombre, si necesita archivo de log, una explicación de su objetivo o un ícono. En la figura 5.25 se muestra la base de datos creada en MongoLab.

The image shows a screenshot of the MongoLab interface. It displays two documents from a collection. The first document has the following structure: "_id": {"\$oid": "5589342a7be1cb130bca9529"}, "numero": "01", "text": "Cifrado", "descripcion": "Comprobar la cantidad de datos de una columna de una tabla de una base de datos se encuentran cifrados.", "info": 0, "log": true, "icon": "glyphicon glyphicon-lock", "v": 0. The second document has a similar structure but with a different ID. The interface includes a header with "records / page" set to 10 and a status bar indicating "[1 - 6 of 6]". There are also standard MongoDB edit and delete icons on the right side of each document.

Figura 5.25 Base de datos de evaluaciones alojada en MongoLab

Otro cambio realizado en esta pestaña es la adición de un botón de información para mostrar una descripción del objetivo de la evaluación, además de un ícono para mejorar la familiaridad con la que el usuario perciba la herramienta. En la figura 5.26 se puede observar como ha quedado la pestaña de evaluaciones tras estos cambios. En esta imagen también se puede ver otros pequeños cambios realizados sobre la interfaz, ya que, si no se sube el archivo de log en la pestaña de configuración las evaluaciones que lo requieran se mostrarán deshabilitadas.

Además también se ha añadido un ícono para que mientras se esté ejecutando una evaluación se indique que dicha evaluación se encuentra en progreso, o que a la finalización de esta se muestre un ícono con un “tick” verde para confirmar que esta ha acabado.



Figura 5.26 Interfaz final pestaña de evaluaciones

Con la interfaz de la imagen anterior creada, el siguiente paso será diferenciar las evaluaciones según las necesidades que tengan, es decir, que al darle al botón de cada evaluación surja una nueva pantalla, en la cual, se pidan los datos necesarios para realizar cada una:

- Evaluación 1, cifrado: será necesario indicar el nombre de la base de datos, de la tabla y finalmente de la columna sobre la cual se quiere comprobar si tiene los registros cifrados.
- Evaluación 2, permisos usuarios: en este caso, habrá que introducir en el sistema el permiso que se desea evaluar.
- Evaluación 3, accesos máximos: para esta evaluación es necesario dar el nombre de la base de datos, así como los datos del usuario a evaluar (nombre de usuario

y contraseña), también hay que indicar el número máximo de conexiones que se le suponen.

- Evaluación 4, usuarios eliminados: para comprobar si los usuarios eliminados ya no disponen de permisos en el sistema, será necesario introducir un archivo que contenga el nombre de los usuarios supuestamente eliminados.
- Evaluación 5, accesos erróneos: en este caso, no es necesario introducir ningún dato más al sistema, ya que, sólo necesita el archivo de log introducido en la pestaña de configuración.
- Evaluación 6, acceso en horario permitido: para esta evaluación será necesario subir un archivo con los usuarios a evaluar y sus horarios permitidos.

En la figura 5.27, se pueden ver algunos ejemplos de cómo se han diseñado estas pantallas para las evaluaciones.

The figure consists of two vertically stacked screenshots of a web application interface. Both screenshots have a header with three tabs: 'Configuración inicial' (disabled), 'Evaluaciones' (selected and highlighted in blue), and 'Informe Final' (disabled).

The top screenshot is titled 'Evaluación 01 - Cifrado'. It contains the following text: 'Comprobar la cantidad de datos de una columna de una tabla de una base de datos se encuentran cifrados.' Below this are three input fields with labels: 'Nombre de la base de datos' (with an empty input field), 'Tabla de la base de datos' (with an empty input field), and 'Columna a evaluar' (with an empty input field). At the bottom are two buttons: 'Volver' and 'Evaluar' (highlighted in blue).

The bottom screenshot is titled 'Evaluación 06 - Acceso en horario permitido'. It contains the following text: 'Comprueba si los usuarios han accedido al sistema en su horario permitido, para ello deberá subir un archivo con los usuarios y sus horas de acceso.' Below this is a file input field labeled 'Lista de usuarios' with a note: 'N Seleccionar archivo hado' and a sub-instruction: 'Suba un archivo con los horarios de los usuarios del sistema, ejemplo: usuario@localhost in: 16:00 || out: 21:00'. At the bottom are two buttons: 'Volver' and 'Evaluar' (highlighted in blue).

Figura 5.27 Ejemplos de pantallas de evaluaciones

Una vez se tienen estas interfaces creadas, es el momento de crear una serie de scripts que ejecuten los pasos necesarios para realizar las evaluaciones. Estos scripts se llamarán desde el frontend de la aplicación, ya que, al pulsar en el botón “Evaluar” de cada una de las evaluaciones se creará una llamada de tipo POST que será ejecutado en el backend de la aplicación.

Algunos de estos scripts suponían la realización de dos trabajos MapReduce, aquellos en los que era necesario usar la herramienta Sqoop para obtener datos de la base de

datos, por ello, se ha considerado oportuno dividirlos en dos scripts, en los cuales, cuando acabe la ejecución del primero de ellos, se llamará al segundo. Esto se hace para evitar que se ejecuten varias llamadas POST producidas por superar el tiempo de espera.

En el listado 5.23 se puede ver un ejemplo de estos scripts creados.

Script 4a

```
#!/usr/bin/python
import os
import sys

os.system("sqoop import --connect jdbc:mysql://" + sys.argv[1] + "/information_schema --table user_privileges --username " + sys.argv[2] +
" --password " + sys.argv[3] + " --target-dir /user/julio/input/EV4/Ev4" + sys.argv[5] + " -m 1");
```

Script 4b

```
os.system("hadoop jar /usr/local/hadoop/share/hadoop/tools/lib/hadoop-streaming-2.6.0.jar -input /user/julio/input/EV4/Ev4" +
"/part-m-00000 -file /home/julio/Escritorio/Interfaz/EvaluacionesTFG/EV4_UsersEliminados/mapper.py -mapper 'python mapper.py' -file
/home/julio/Escritorio/Interfaz/EvaluacionesTFG/EV4_UsersEliminados/reducer.py -reducer 'python reducer.py " + sys.argv[4] +
"" -output /user/julio/output/EV4/Ev4" + sys.argv[5]);"
os.system("hadoop fs -copyToLocal /user/julio/output/EV4/Ev4" + sys.argv[5] + "/part-00000 ./resultados/")
os.system("mv ./resultados/part-00000 ./resultados/EV4" + sys.argv[5])
os.system("hadoop fs -rm -r /user/julio/output/EV4/Ev4" + sys.argv[5])
os.system("hadoop fs -rm -r /user/julio/input/EV4/Ev4" + sys.argv[5])
```

Listado 5.23 Ejemplo de los scripts de la evaluación 4

Como se puede ver, estos scripts están implementados en python y para ejecutarlos se sigue el siguiente flujo de datos:

1. Pulsar el botón evaluar en la pantalla de la evaluación.
2. El HTML llamará al método que realizará la evaluación.
3. Dicho método ejecutará un POST sobre la ruta de la evaluación a realizar.
4. El backend de la aplicación será el encargado de gestionar esta ruta, y ejecutar un script, tras el cual, devolverá el resultado generado.
5. En caso de que la evaluación requiera de utilizar la herramienta Sqoop para obtener datos de la base de datos, será necesario utilizar dos rutas: una para la recolección de dichos datos y otra para realizar la tarea MapReduce con la que se obtendrá el resultado. Para realizar esto, el primer POST hecho en el frontend, en caso de tener éxito ejecutará un segundo POST, el cual, ya sí recogerá el resultado de la evaluación.
6. En caso de éxito, el método recogerá la ruta del archivo de resultado generado tras realizar la evaluación.

Este proceso se puede ver resumido en la figura 5.28, en la cual, se toma como ejemplo el flujo necesario para realizar la evaluación 4 de la herramienta (usuarios eliminados).

```

<button type="button" class="btn btn-primary" ng-click="evaluacion4()>
    Evaluar
</button>

$scope.evaluacion4 = function(){
    volver();
    loading[3] = true;
    res_ev4.contador += 1;
    $scope.formConf.contador = res_ev4.contador;
    $scope.formConf.archivo_aux = res_ev4.archivo_aux;

    $http.post('/api/ev4', $scope.formConf)
        .success(function(data) {
            $http.post('/api/ev4b', $scope.formConf)
                .success(function(data) {
                    loading[3] = false;
                    done[3] = true;
                    if (res_ev4.resultado != "") {
                        res_ev4.resultado = res_ev4.resultado + data + "||";
                    } else {
                        res_ev4.resultado = data;
                    }
                })
                .error(function(data) {
                    console.log('Error: ' + data);
                });
        })
        .error(function(data) {
            console.log('Error: ' + data);
        });
};

app.post('/api/ev4', function(req, res){
    var spawn = require('child_process').spawn,
        pythonProcess = spawn('python', ['./scripts/scriptEV4.py', req.body.bd, req.body.user, req.body.pass, req.body.archivo_aux,
        pythonProcess.stdout.on('data', function(data){
            console.log(' ' + data);
        });
        pythonProcess.stderr.on('data', function(data){
            console.log(' ' + data);
        });
        pythonProcess.on('close', function(code){
            console.log('child process exited with code: ' + code);
            res.json("OK");
        });
    });
});

app.post('/api/ev4b', function(req, res){
    var spawn = require('child_process').spawn,
        pythonProcess = spawn("python", ['./scripts/scriptEV4b.py', req.body.bd, req.body.user, req.body.pass, req.body.archivo_aux,
        pythonProcess.stdout.on('data', function(data){
            console.log(' ' + data);
        });
        pythonProcess.stderr.on('data', function(data){
            console.log(' ' + data);
        });
        pythonProcess.on('close', function(code){
            console.log('child process exited with code: ' + code);
            res.json({_dirname:"/resultados/EV4"+(req.body.contador).toString()});
        });
    });
});

```

Tras pulsar en el botón de evaluar, se llama a la función evaluación 4.

Esta hará un POST a ev4, la cual, será la encargada de realizar el Sqoop de la evaluación. Una vez realizado, el flujo volverá a la función evaluación 4, que hará otro POST, en este caso a ev4b, donde se completará la evaluación.

Figura 5.28 Flujo de datos para la ejecución de un script

Finalmente, cabe destacar que en estos scripts se incluye también la funcionalidad de eliminar los datos de entrada para no conservar ningún dato en el sistema, así como crear una copia de los archivos de resultado desde el sistema Hadoop hasta el servidor.

5.9.5 Revisión del Sprint

A la finalización de este sprint, se ha conseguido el octavo incremento de la herramienta, en el cual, se ha realizado la implementación de la herramienta web que integra todas las funcionalidades del sistema, permitiendo la realización de todas las evaluaciones en una misma plataforma.

Para ello, primero se ha diseñado e implementado una interfaz sin funcionalidad, para

luego posteriormente implementar las distintas evaluaciones de las que dispone el sistema. Ha sido necesario la creación de distintos scripts para lograrlo.

En cuanto a la estimación temporal realizada para este sprint, se ha quedado corta y por tanto, ha sido necesario la realización de horas extras para evitar que la planificación temporal se viera trastocada. Esto probablemente ha sido fruto de la poca experiencia del equipo de desarrollo con tecnologías de este tipo.

5.10 Sprint 9

El sprint 9 será el sprint final de la herramienta, y su objetivo será representar los resultados obtenidos en las diferentes evaluaciones del sistema. Para ello, habrá que estudiar alguna tecnología capaz de proporcionar gráficos y habrá que integrarla en la herramienta. Con esto, se le dará funcionalidad a la tercera pestaña de la herramienta: informe final.

5.10.1 Refinamiento de la Pila del Producto

La Pila del Producto no ha sufrido modificaciones tras su revisión.

5.10.2 Planificación del Sprint

Este sprint realizará dos historias de usuario: la historia de usuario 8 (“Quiero obtener unos gráficos para representar el informe de resultados.”) y la historia de usuario 15 (“Quiero obtener un pdf del informe con los resultados de la evaluación de seguridad.”).

La historia de usuario 8 tiene como objetivo la presentación de los resultados obtenidos de la ejecución de las evaluaciones en forma de gráficos, para ello, habrá que decidir qué tecnología permite realizarlo y adaptarlo a la herramienta.

Por otro lado, la historia de usuario 15 surge con el objetivo de dar la posibilidad al auditor que haga uso de la herramienta de descargar un pdf con el informe final de seguridad derivado de las evaluaciones realizadas y con los gráficos de la anterior historia de usuario.

Finalmente, la pila de sprint está formada dos historias de usuario que son las siguientes:

Historia de Usuario	
Número: 8	
Nombre historia: Representar los resultados con gráficos	
Valor de negocio: Alto	Riesgo en desarrollo: Alto
Esfuerzo: 10,5 horas	Sprint asignado: 9
Programador responsable: Julio Moreno García-Nieto	
Descripción: Dotar a la herramienta con la capacidad de representar mediante gráficos los resultados obtenidos de ejecutar las distintas evaluaciones.	
Postcondición: Gráficos para los resultados en la pestaña de informe final de la aplicación web.	
Tareas: <ul style="list-style-type: none"> Decidir tecnología a usar para mostrar los gráficos. Decidir qué gráficos se adapta a cada evaluación. Implementar los gráficos. 	

Tabla 5.20 Historia de usuario "Representar los resultados con gráficos"

Historia de Usuario	
Número: 15	
Nombre historia: Obtener pdf del informe final	
Valor de negocio: Bajo	Riesgo en desarrollo: Bajo
Esfuerzo: 10,5 horas	Sprint asignado: 9
Programador responsable: Julio Moreno García-Nieto	
Descripción: Dar a la herramienta la posibilidad de generar un pdf con los datos y gráficos del informe final.	
Postcondición: Generación de pdf con el informe final.	
Tareas: <ul style="list-style-type: none"> Generar un pdf que tenga el contenido de la pestaña informe final de la herramienta. 	

Tabla 5.21 Historia de usuario "Obtener pdf del informe final"

5.10.3 Desarrollo de historia de usuario: *Representar los resultados con gráficos*

Para el desarrollo de esta historia de usuario, lo primero que se ha establecido en planificación es decidir cuál será la tecnología usada para representar los gráficos requeridos. En este caso, se ha decidido hacer uso de la tecnología Google Charts, ya que, permite la realización de una serie de gráficos con los cuales se ha determinado que es posible representar los resultados. Para ello, ha sido necesario instalar un módulo denominado Angular Google Charts.

Una vez determinada la tecnología a usar, el siguiente paso es determinar qué gráfico se usará en cada caso, para ello se ha realizado la siguiente tabla:

Evaluación	Resultados	Gráfico usado
Evaluación 1: Cifrado	<ul style="list-style-type: none"> • Cifrados. • No cifrados. 	Sectores
Evaluación 2: Permisos usuario	<ul style="list-style-type: none"> • Con permiso. • Sin permiso. 	Sectores
Evaluación 3: Accesos máximos	<ul style="list-style-type: none"> • Cumple o no cumple. 	--
Evaluación 4: Usuarios eliminados	<ul style="list-style-type: none"> • Usuarios eliminados con permisos. • Usuarios eliminados sin permisos. 	Sectores
Evaluación 5: Accesos erróneos	<ul style="list-style-type: none"> • Número de accesos erróneos por usuario. 	Barras
Evaluación 6: Acceso en horario permitido	<ul style="list-style-type: none"> • Accesos dentro de horario permitido. • Accesos fuera de horario permitido. • Accesos de usuarios no encontrados. 	Sectores

Tabla 5.22 Gráficos a utilizar en función de la evaluación

Como se puede observar en la mayoría de los casos se usará gráficos de sectores, menos en el caso de la evaluación 3, que al evaluar sólo una condición no tiene sentido representarla con gráfico sino que se usará un ícono para indicar si se cumple o no, y la evaluación 5, la cual, tiene como resultado una lista con el número de accesos erróneos por usuario, y por tanto, se considera que se representará los resultados de una forma más clara usando un gráfico de barras.

Finalmente, para completar la historia de usuario sólo queda implementar en el sistema los gráficos, para ello, se creará un gráfico por evaluación, el cual, se mostrará o no en función de si se ha realizado la evaluación.

En caso positivo, se modificarán los datos propios del gráfico para introducir los datos obtenidos de los archivos resultado de cada evaluación. Por ejemplo, en el caso de la segunda evaluación (permisos usuario) se obtiene un archivo de resultados, en el cual, se indica en dos líneas cuántos usuarios tienen dicho permiso y cuántos no, por tanto, para representar esos datos habrá que abrir el archivo de resultados y transformar dichas cifras para que sean legibles por el gráfico.

Como resultado de este proceso, se obtiene lo que se puede ver figura 5.29.

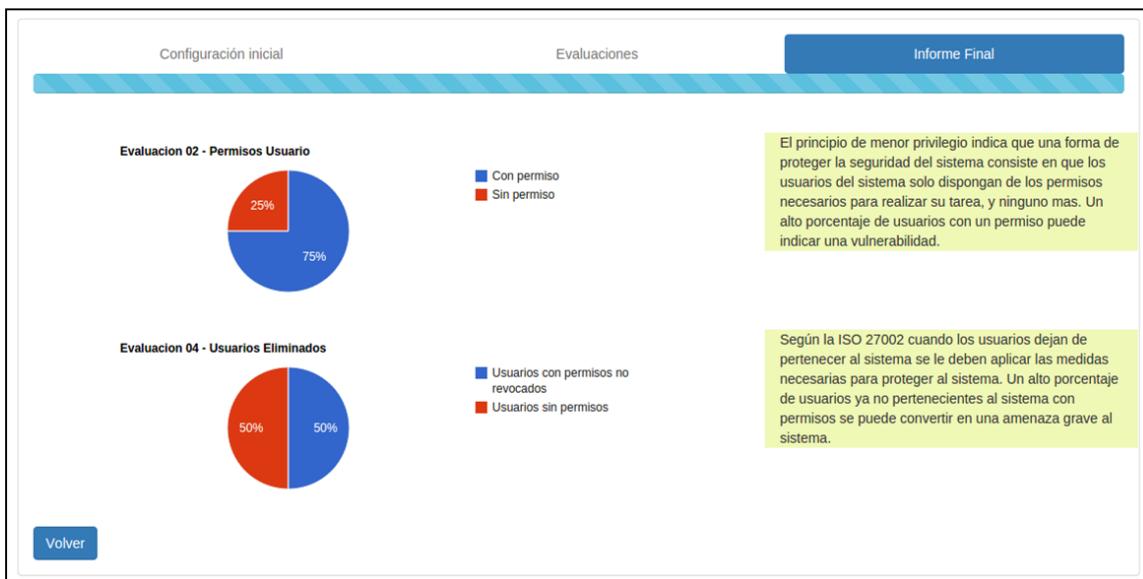


Figura 5.29 Informe final de la herramienta

Como se puede ver en la imagen anterior, para completar la información dada por el informe final se ha añadido un pequeño comentario para cada evaluación para ayudar al auditor a estudiar los resultados obtenidos.

5.10.4 Desarrollo de historia de usuario: *Obtener pdf del informe final*

Durante el desarrollo de esta historia de usuario, lo que se busca es plasmar en un pdf en el informe final obtenido en la anterior historia de usuario. Para ello, ha sido necesario añadir un nuevo botón a la interfaz “Crear PDF”, el cual, llama a una función que se encarga de guardar en una variable el código HTML de los gráficos que se representan.

Para conseguirlo, es necesario añadir un id a cada una de las div que sirven para representar los gráficos, de forma que cuando esa evaluación se realiza el gráfico también se plasmará en el PDF. Este método se puede ver en el listado 5.24.

Una vez se tienen los datos que se quieren imprimir en el PDF, como se puede ver en el listado anterior, se crea una nueva ventana en la cual se escriben dichos datos. Finalmente, el resultado de ejecutar el método es una ventana en la cual se permite generar un PDF como el que puede observarse en la figura 5.30.

```

$scope.createPDF = function(){
    var imprimir = "<h1>Informe Final</h1>" 
    if (vista.chart_1 == true)
        var imprimir = imprimir + document.getElementById('informeFinal1').innerHTML;
    if (vista.chart_2 == true)
        var imprimir = imprimir + document.getElementById('informeFinal2').innerHTML;
    if (vista.chart_3 == true)
        var imprimir = imprimir + document.getElementById('informeFinal3').innerHTML;
    if (vista.chart_4 == true)
        var imprimir = imprimir + document.getElementById('informeFinal4').innerHTML;
    if (vista.chart_5 == true)
        var imprimir = imprimir + document.getElementById('informeFinal5').innerHTML;
    if (vista.chart_6 == true)
        var imprimir = imprimir + document.getElementById('informeFinal6').innerHTML;
    var myWindow = window.open('', '', 'width=1024', 'height=800');
    myWindow.document.write(imprimir);
    myWindow.print();
}

```

Listado 5.24 Método para crear el PDF

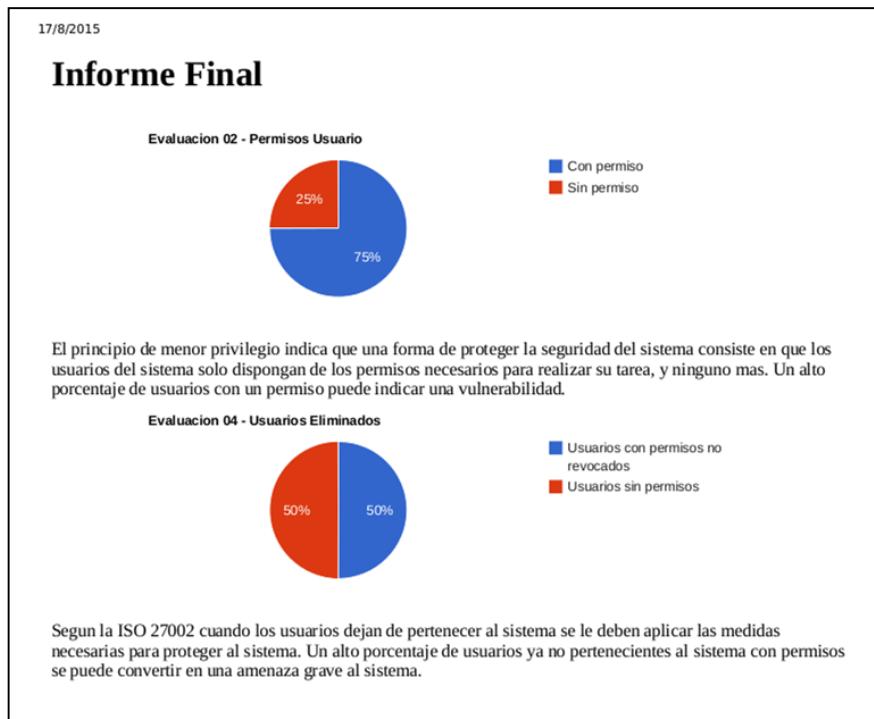


Figura 5.30 Informe final en PDF

5.10.5 Revisión del Sprint

A la finalización de este sprint, se consigue el noveno y último incremento de la herramienta, con el cual, se considera finalizado el alcance de este TFG. En este sprint, se ha conseguido representar los resultados obtenidos en las evaluaciones de forma que

conformen el informe final de seguridad.

Estos resultados han sido expresados mediante gráficos de distinto tipo en función de la evaluación, y se verán acompañados de un comentario que ayude al auditor a interpretar dichos resultados. Además se ha añadido la posibilidad de generar este informe final en pdf para que el usuario pueda guardarlo y utilizarlo cuando considere oportuno.

Sobre la estimación temporal realizada para este sprint es necesario decir que en el caso de la primera historia de usuario se ha quedado un poco corta, pero por otro lado, la segunda historia de usuario se ha resuelto de una forma más rápida de lo esperado, por tanto, la estimación ha quedado bastante cerca del esfuerzo real necesario.

6 Conclusiones y propuestas

Durante este capítulo se determina si se han cumplido los objetivos marcados durante el capítulo 2 de la presente memoria. Además se establecerán unas propuestas de trabajo futuro para mejorar la herramienta. Finalmente, se realiza una opinión personal del autor del proyecto tras la finalización del mismo.

6.1 Análisis de consecución de los objetivos del proyecto

El objetivo principal del TFG que se marcó en el capítulo 2 era el siguiente:

El objetivo principal del TFG será la realización de una herramienta web que permita la evaluación de la seguridad de una base de datos relacional mediante el uso de técnicas de Big Data.

Este objetivo ha sido cumplido completamente con el desarrollo de la herramienta para la evaluación de la seguridad en bases de datos relacionales, cuyo desarrollo se ha documentado con esta memoria. Además del objetivo principal, se marcaron una serie de objetivos parciales, que se pueden ver resumidos en la siguiente tabla:

Id Objetivo	Objetivo	Consecución
O.1	Estudiar los problemas típicos de seguridad de las bases de datos relacionales. Una vez conocidos, elección de las evaluaciones a realizar por la herramienta.	
O.2	Introducir la información de una base de datos relacional en un sistema Big Data.	
O.3	Implementación de las evaluaciones seleccionadas mediante técnicas de Big Data (en las que sean necesario).	
O.4	Realización de una interfaz web que permita realizar todas las evaluaciones ya implementadas.	
O.5	Generación de un informe final con los resultados obtenidos de ejecutar las evaluaciones.	

Tabla 6.1 Consecución de objetivos parciales

- El objetivo 1 se cumple gracias al estudio necesario para realizar el estado de la cuestión resumida en el capítulo 3. Gracias a esta información se eligieron las evaluaciones a realizar durante el sprint 0.
- El objetivo 2 se cumple durante el desarrollo del sprint 1.
- El objetivo 3 es cumplido durante el desarrollo de los sprints entre 2 y 7.
- El objetivo 4 se alcanza durante el desarrollo del sprint 8.
- El objetivo 5 se cumple durante el desarrollo del sprint 9.

Por tanto, se puede concluir que se han cumplido todos los objetivos especificados en el capítulo 2.

6.2 Propuestas de trabajos futuros

Durante el desarrollo del TFG se pensaron diferentes propuestas para mejorar la herramienta, que debido a diversos factores, no han sido implementadas a la finalización del TFG, en la siguiente lista se comentan las más importantes:

- Implementación de nuevas evaluaciones de seguridad que complementen la funcionalidad de la herramienta.
- Desplegar el sistema de Big Data en varios nodos para mejorar la eficiencia de la herramienta.
- Desarrollar un módulo nuevo de gestión de usuarios en la herramienta, que permita que cada usuario tenga sus informes finales almacenados.

6.3 Otros datos de interés

6.3.1 Participación en eventos de Seguridad y Big Data

Durante el desarrollo del TFG se realizaron una serie de actividades que favorecieron el entendimiento de los temas de seguridad informática y Big Data:

- Participación en: “2nd ISO/IEC – JTC 1/WG9 for Big Data Meeting” para la creación de una norma ISO de Big Data, celebrada en Ciudad Real durante el mes de julio de 2015.
- Participación en el curso de verano de doctorando “Diseño Seguro y Análisis de Amenazas en Sistemas”, celebrado en León durante el mes de julio de 2015.

6.3.2 Publicaciones

- Realización de la publicación: “Towards a Security Model for Big Data” de David G. Rosado, Ismael Caballero, Julio Moreno, Manuel Ángel Serrano y Eduardo Fernandez-Medina; **aceptado para publicación** en la conferencia CIBSI/TIBETS 2015 [75].

6.4 Opinión personal

El desarrollo de este TFG me ha permitido comprender de una forma más profunda y práctica temas cursados durante la carrera, además de adquirir una gran cantidad de conocimientos tanto tecnológicos (Big Data, AngularJS, NodeJS, etc.) como teóricos relacionados con la seguridad de la información, los cuales, en un principio me resultaban ajenos en mayor o menor grado.

Así, me gustaría destacar que estos conocimientos adquiridos no caerán en saco roto sino que me han servido para fijarme un objetivo de futuro profesional, y es que dentro de las distintas ramas de la informática, la seguridad ha despertado mi curiosidad e interés sobre el tema.

La finalización del TFG supone el fin de una etapa de mi vida, pero lo expuesto en el anterior párrafo hace que sea imposible que este sea el punto y final a mi formación académica sino que significará un “continuará”.

Julio Moreno García-Nieto

Ciudad Real, a 25 de Agosto de 2015

Bibliografía

- [1] Mireia Delandes Palomares, «Historia de las bases de datos - culturainformatica.es», 04-nov-2011. .
- [2] O. Pons, *Introducción a los sistemas de bases de datos*, Edición: 1. Madrid: Ediciones Paraninfo, S.A, 2008.
- [3] D. Gurnpreet, *Information Security Management: Global Challenges in the New Millennium: Global Challenges in the New Millennium*. Idea Group Inc (IGI), 2000.
- [4] «ISO 27000.» .
- [5] «MONOGRÁFICO: Introducción a la seguridad informática - Seguridad de la información / Seguridad informática | Observatorio Tecnológico.» [En línea]. Disponible en:
<http://recursostic.educacion.es/observatorio/web/es/component/content/article/1040-introduccion-a-la-seguridad-informatica?start=1>. [Accedido: 31-jul-2015].
- [6] David García Rosado, «Fundamentos de Seguridad.» .
- [7] C. Vergara, «Ataque a la base de datos: ataque y seguridad a la base de datos.», *ATAQUE A LA BASE DE DATOS*, 14-jul-2012. .
- [8] «CTT - MAGERIT versión 3», 09-mar-2010. [En línea]. Disponible en:
<http://administracionelectronica.gob.es/ctt/magerit/descargas#.VdxBJfnQOug>. [Accedido: 25-agosto-2015].
- [9] ISACA, «COBIT 5 - Un marco de negocio para el gobierno y la gestión de las TI de la empresa.» .
- [10] Emmie Zuberfly, «Advantages and disadvantages of DBMS & different data types in MS Access.» .
- [11] «Too Much Content: A World of Exponential Information Growth», *The Huffington Post*. [En línea]. Disponible en: http://www.huffingtonpost.com/brett-king/too-much-content-a-world-_b_809677.html. [Accedido: 03-agosto-2015].
- [12] V. Mayer-Schonberger y K. N. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Boston: Houghton Mifflin, 2013.
- [13] ISO/IEC, «Big Data - Preliminary Report», 2014.
- [14] «Big Data, Better Marketing | Sweetly Social.» .
- [15] Daniel J. Ollero, «Claves para entender el Big Data», *ELMUNDO*, 31-dic-2014. [En línea]. Disponible en:
<http://www.elmundo.es/economia/2014/12/31/54a2f77622601dd2418b456c.html>. [Accedido: 19-mayo-2015].
- [16] D. S. Herrmann, *Using the Common Criteria for IT Security Evaluation*. CRC Press, 2002.
- [17] M. Whitman y H. Mattord, *Principles of Information Security*. Cengage Learning, 2011.
- [18] Carlos Villarrubia Jiménez, «Adopción de pautas de seguridad informática.» .
- [19] K. A. Paul van Kessel, «Cyber security - EY-GISS-Under-cyber-attack.pdf.» .
- [20] «ISO27000.es - El portal de ISO 27001 en español. Gestión de Seguridad de la Información.» [En línea]. Disponible en: <http://www.iso27000.es/iso27000.html>. [Accedido: 15-agosto-2015].
- [21] ISO/IEC, «ISO 27002:2013 - Information technology, security techniques, code

- of practise for information security controls», oct. 2013.
- [22] Jorge Domínguez Chávez, «Principios básicos de seguridad en bases de datos», jul. 2015.
- [23] Luis Alonso Romero, «Seguridad Informática. Conceptos generales.» [En línea]. Disponible en:
<http://campus.usal.es/~derinfo/Activ/Jorn02/Pon2002/LARyALSL.pdf>. [Accedido: 16-agosto-2015].
- [24] «Databases—Vulnerabilities, Costs of Data Breaches and Countermeasures», *InfoSec Institute*. [En línea]. Disponible en:
<http://resources.infosecinstitute.com/databases-vulnerabilities-costs-of-data-breaches-and-countermeasures/>. [Accedido: 16-agosto-2015].
- [25] M. Luna, «Criptografia: Criptografía y los Diferentes Tipos de Cifrados», *Criptografía*, 06-oct-2012. .
- [26] E. por S. D. Luz, «Criptografía : Algoritmos de cifrado de clave simétrica», *RedesZone*. [En línea]. Disponible en:
<http://www.redeszone.net/2010/11/04/criptografia-algoritmos-de-cifrado-de-clave-simetrica/>. [Accedido: 16-agosto-2015].
- [27] E. por S. D. Luz, «Criptografía : Algoritmos de cifrado de clave asimétrica», *RedesZone*. [En línea]. Disponible en:
<http://www.redeszone.net/2010/11/16/criptografia-algoritmos-de-cifrado-de-clave-asimetrica/>. [Accedido: 16-agosto-2015].
- [28] M. G. P. V. Eduardo Fernández-Medina Patón, «Metodología para el Diseño de Bases de Datos Seguras.», UCLM.
- [29] «Home | Scrum Guides.» [En línea]. Disponible en:
<http://www.scrumguides.org/>. [Accedido: 25-agosto-2015].
- [30] «Introduction to Test Driven Development (TDD).» [En línea]. Disponible en:
<http://agiledata.org/essays/tdd.html>. [Accedido: 30-junio-2015].
- [31] K. Schwaber, *Agile Project Management with Scrum*. Microsoft Press, 2004.
- [32] «Scrum Sprint.» .
- [33] John Drew, «Scrum Components.» .
- [34] Mountain Goat Software, «Sprint Backlog and the Scrum Sprint.» [En línea]. Disponible en: <https://www.mountaingoatsoftware.com/agile/scrum/sprint-backlog>. [Accedido: 25-mayo-2015].
- [35] G. L. Villarreal, «Notas de Scrum», *Linux LPMagazine*, 2008.
- [36] «Introducción a TDD (Test Driven Development) | lorddudun Blog.» .
- [37] «Advantages of Test-Driven Development | Agile Pain Relief.» [En línea]. Disponible en: <https://agilepainrelief.com/notesfromatooluser/2008/10/advantages-of-tdd.html#.VcCSibfPFGY>. [Accedido: 04-agosto-2015].
- [38] «A Brief Introduction to Kanban | The Agile Coach.» [En línea]. Disponible en:
<https://www.atlassian.com/agile/kanban>. [Accedido: 30-junio-2015].
- [39] «Software Design Tools for Agile Teams, with UML, BPMN and More.» [En línea]. Disponible en: <http://www.visual-paradigm.com/>. [Accedido: 26-mayo-2015].
- [40] «Balsamiq Mockups - Balsamiq.» [En línea]. Disponible en:
<https://balsamiq.com/products/mockups/>. [Accedido: 30-junio-2015].
- [41] «GanttProject: free desktop project management app.» [En línea]. Disponible en:
<http://www.ganttproject.biz/>. [Accedido: 26-mayo-2015].
- [42] «Cómo utilizar Trello para organizar tu vida casi al completo», *Gizmodo en*

- Español.* [En línea]. Disponible en: <http://es.gizmodo.com/como-organizar-toda-tu-vida-utilizando-trello-1684529913>. [Accedido: 26-may-2015].
- [43] «Trello.» [En línea]. Disponible en: <https://trello.com>. [Accedido: 26-may-2015].
- [44] «Hatjitsu :: Online Scrum Planning Poker for Agile Projects.» [En línea]. Disponible en: <http://hat.jit.su/>. [Accedido: 28-may-2015].
- [45] «Apache™ Hadoop®!» [En línea]. Disponible en: <https://hadoop.apache.org/>. [Accedido: 26-may-2015].
- [46] «Apache Hadoop | MapR.» [En línea]. Disponible en: <https://www.mapr.com/products/apache-hadoop>. [Accedido: 25-ago-2015].
- [47] «Welcome to Apache Pig!» [En línea]. Disponible en: <https://pig.apache.org/>. [Accedido: 26-may-2015].
- [48] M. Lutz, *Learning Python*, Edición: 5. Beijing: O'Reilly Media, 2013.
- [49] «Installation and quick start — nose 1.3.7 documentation.» [En línea]. Disponible en: <https://nose.readthedocs.org/en/latest/>. [Accedido: 30-jun-2015].
- [50] «Introducción a Node.js - Rafa Muñoz.» [En línea]. Disponible en: <http://www.rmunoz.net/introduccion-a-node-js.html>. [Accedido: 30-jun-2015].
- [51] «Learn AngularJS», *Codecademy*. [En línea]. Disponible en: <http://www.codecademy.com/es/learn/learn-angularjs>. [Accedido: 28-may-2015].
- [52] «JSON.» [En línea]. Disponible en: <http://json.org/>. [Accedido: 26-may-2015].
- [53] «Bootstrap 3, el manual oficial.» [En línea]. Disponible en: https://librosweb.es/libro/bootstrap_3/. [Accedido: 28-may-2015].
- [54] «¿Qué es Bootstrap?», *OpenWebCMS* .
- [55] «¿Qué es HTML?» [En línea]. Disponible en: <http://es.html.net/tutorials/html/lesson2.php>. [Accedido: 28-may-2015].
- [56] «W3C HTML.» [En línea]. Disponible en: <http://www.w3.org/html/>. [Accedido: 28-may-2015].
- [57] «Guía Breve de CSS.» [En línea]. Disponible en: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>. [Accedido: 28-may-2015].
- [58] «Sublime Text: The text editor you'll fall in love with.» [En línea]. Disponible en: <http://www.sublimetext.com/>. [Accedido: 28-may-2015].
- [59] «Guía de Sublime Text: ¿El mejor editor de código? | Emezeta.» [En línea]. Disponible en: <http://www.emezeta.com/articulos/guia-sublime-text>. [Accedido: 28-may-2015].
- [60] «Using Google Charts», *Google Developers*. [En línea]. Disponible en: <https://developers.google.com/chart/interactive/docs/>. [Accedido: 18-jul-2015].
- [61] «MySQL :: MySQL 5.7 Reference Manual :: 1.3.1 What is MySQL?» [En línea]. Disponible en: <http://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>. [Accedido: 28-may-2015].
- [62] «¿Qué es MySQL? :: esepestudio, especialistas web.» [En línea]. Disponible en: <http://www.esepestudio.com/noticias/que-es-mysql>. [Accedido: 25-ago-2015].
- [63] «Sqoop -.» [En línea]. Disponible en: <http://sqoop.apache.org/>. [Accedido: 28-may-2015].
- [64] K. Chodorow, *MongoDB: The Definitive Guide*. O'Reilly Media, Inc., 2013.
- [65] «Software de procesamiento de texto y documentos | Microsoft Word.» [En línea]. Disponible en: <https://products.office.com/es-es/word>. [Accedido: 02-jun-

2015].

- [66] «Zotero | Bibliotecas Universidad de Salamanca.» [En línea]. Disponible en: <http://bibliotecas.usal.es/zotero>. [Accedido: 02-jun-2015].
- [67] «GIMP Descargas, tutoriales y foros. Alternativa a Photoshop gratis y libre.» [En línea]. Disponible en: <http://www.gimp.org.es/>. [Accedido: 18-jul-2015].
- [68] Dr.Caos, «¿Qué es Virtualbox y como usar Virtualbox?», *Dr.Caos / Internet Desde otro Punto de vista.* .
- [69] «DESARROLLO Y GESTION DE PROYECTOS INFORMATICOS - STEVE MCCONNELL, comprar el libro.» [En línea]. Disponible en: <http://www.casadellibro.com/libro-desarrollo-y-gestion-de-proyectos-informaticos/9788448112295/577824>. [Accedido: 18-jul-2015].
- [70] «MySQL :: MySQL Applier for Hadoop.» [En línea]. Disponible en: <https://dev.mysql.com/tech-resources/articles/mysql-hadoop-applier.html>. [Accedido: 05-ago-2015].
- [71] R. D. L. Rosa, «Script Inside: Listado de palabras en castellano», *Script Inside*, 27-ene-2012..
- [72] «What is MapReduce? - Definition from WhatIs.com», *SearchCloudComputing*. [En línea]. Disponible en: <http://searchcloudcomputing.techtarget.com/definition/MapReduce>. [Accedido: 06-ago-2015].
- [73] «MySQL :: MySQL 5.0 Reference Manual :: 19 INFORMATION_SCHEMA Tables.» [En línea]. Disponible en: <https://dev.mysql.com/doc/refman/5.0/en/information-schema.html>. [Accedido: 06-ago-2015].
- [74] «Tutorial de AngularJS. Ejemplo de aplicación web y API REST con NodeJS», *Carlos Azaustre / Desarrollador y Formador Web*. [En línea]. Disponible en: <http://carlosazaustre.es/blog/tutorial-ejemplo-de-aplicacion-web-con-angular-js-y-api-rest-con-node/>. [Accedido: 10-ago-2015].
- [75] «:::CIBSI 2015»: [En línea]. Disponible en: <http://cibsi.espe.edu.ec/>. [Accedido: 25-ago-2015].
- [76] edgar R. to edgar, «How to Install Nodejs & Npm on Ubuntu using PPA», *TecAdmin.net*. .

Anexo A. Manual de usuario

Durante este anexo se realizará un manual de uso para la herramienta web desarrollada durante el TFG, para ello, este manual se estructurará en las tres pestañas en las que se divide la herramienta, incluyendo todas las capturas necesarias para su mejor comprensión.

Para la realización de este manual se ha utilizado la base de datos de los presupuestos de la Comunidad de Madrid de acceso público y ya comentada con anterioridad durante este TFG. Para mostrar unos resultados interesantes se han añadido una serie de usuarios a la base de datos y se han modificado algunos registros con este fin, como por ejemplo se han cifrado algunos de los registros para la evaluación 1.

1. Configuración

En la pestaña de configuración se muestran los datos fundamentales para la realización de las evaluaciones de la herramienta, también permite subir un archivo de log del sistema, necesario para poder realizar las evaluaciones 5 y 6, en caso de no proporcionar dicho archivo de log esas dos evaluaciones permanecerán deshabilitadas hasta que se suba. En la figura A.1 se puede ver la pestaña inicial de configuración.

La captura de pantalla muestra la interfaz de usuario para la configuración inicial. El encabezado dice 'Evaluación de seguridad en Bases de datos'. Hay tres pestañas: 'Configuración inicial' (selecciónada), 'Evaluaciones', y 'Informe Final'. Los campos de configuración son:

- Dirección base de datos: ej. localhost
- Usuario "root": ej. root
- Contraseña "root": (campo vacío)
- Archivo de log (opcional): Seleccionar archivo Ningún archivo seleccionado. Sino sube el archivo de log no podrá realizar algunas evaluaciones.

Un botón 'Siguiente' está en la parte inferior izquierda.

Figura A.1 Pestaña inicial de configuración

Para completar esta primera pestaña de configuración, se necesitan los datos de dirección del sistema gestor de bases de datos, nombre de usuario root y contraseña de dicho usuario, además del archivo de log si se desea. En este caso, quedaría de la siguiente forma.

Evaluación de seguridad en Bases de datos

Configuración inicial Evaluaciones Informe Final

Dirección base de datos: localhost

Usuario "root": root

Contraseña "root": ****

Archivo de log (opcional): Seleccionar archivo mysql.log
Sino sube el archivo de log no podrá realizar algunas evaluaciones

Siguiente



Figura A.2 Pestaña configuración completada

Una vez completada esta pestaña, se pulsa en el botón siguiente que lleva a la pestaña de evaluaciones.

2. Evaluaciones

En la pestaña de evaluaciones se puede observar una lista con las evaluaciones a realizar, junto a un botón que proporciona información sobre dicha evaluación. Como se comentó en el anterior apartado, sino se añade el archivo de log las evaluaciones 5 y 6 se verán deshabilitadas, como se puede ver en la siguiente figura.

	03 - Accesos máximos	+Info
	04 - Usuarios eliminados	+Info
	05 - Accesos erróneos	+Info
	06 - Acceso en horario permitido	+Info

Figura A.3 Evaluaciones deshabilitadas sin archivo de log

En caso contrario, se observarán las evaluaciones como se ve en la figura A.4. Cada una de las evaluaciones tiene a su vez su propia pantalla, en la que introducir los datos necesarios para realizar la evaluación.

The screenshot shows a user interface with a top navigation bar containing 'Configuración inicial', 'Evaluaciones' (which is highlighted in blue), and 'Informe Final'. Below this is a list of six evaluation items, each with a small icon and a link to 'Info':

- 01 - Cifrado
- 02 - Permisos usuarios
- 03 - Accesos máximos
- 04 - Usuarios eliminados
- 05 - Accesos erróneos
- 06 - Acceso en horario permitido

At the bottom left are 'Volver' and 'Siguiente' buttons.

Figura A.4 Listado de evaluaciones

a. Evaluación 1

En el caso de la evaluación 1 dedicada al estudio del cifrado de los datos, será necesario introducir los datos referidos a la columna que se quiere evaluar, por tanto, también habrá que añadir el nombre de la tabla donde se encuentra dicha columna y de la base de datos donde, a su vez, se encuentra la tabla.

En este caso concreto se va a evaluar la columna descripción_centro de la tabla presupuestos de la base de datos datos_madrid.

The screenshot shows the configuration for 'Evaluación 01 - Cifrado'. The title 'Evaluación 01 - Cifrado' is at the top. Below it is a descriptive text: 'Comprobar la cantidad de datos de una columna de una tabla de una base de datos se encuentran cifrados.' There are three input fields:

- Nombre de la base de datos: datos_madrid
- Tabla de la base de datos: presupuestos
- Columna a evaluar: descripción_centro

At the bottom left are 'Volver' and 'Evaluar' buttons.

Figura A.5 Evaluación 1

Mientras las evaluaciones se encuentran realizándose se muestra una pequeña imagen para indicar que la evaluación se está procesando. Una vez realizada, esa imagen cambia por un tick verde que indica que se ha finalizado con éxito.

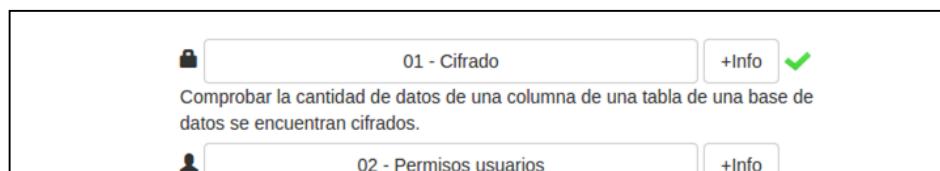


Figura A.6 Evaluación finalizada con éxito

b. Evaluación 2

Para la evaluación dedicada a estudiar los permisos de los usuarios del sistema, es necesario introducir el permiso a evaluar.

Evaluación 02 - Permisos usuarios

Comprobar qué cantidad de usuarios de la base de datos disponen de un permiso concreto.

Permiso a evaluar

Volver **Evaluar**

Figura A.7 Evaluación 2

En este caso, se va a estudiar el porcentaje de usuarios que tienen el permiso “DELETE”.

c. Evaluación 3

Para la evaluación 3 que evalúa si la cantidad de accesos a la base de datos por hora es correcto, será necesario introducir el nombre de la base de datos, los datos de acceso del usuario (nombre y contraseña) y el número de accesos que se le suponen.

Figura A.8 Evaluación 3

En este caso, se estudiará si el usuario “julio” puede realizar 5 conexiones.

d. Evaluación 4

Durante la evaluación 4 se buscará dentro de los permisos de todos los usuarios, si existen usuarios que ya no se encuentran en el sistema pero siguen disponiendo de permisos. Para realizar esta evaluación, será necesario introducir un archivo con los nombres de los usuarios eliminados que se van a evaluar.

Figura A.9 Evaluación 4

Para este caso, se ha utilizado un archivo como el siguiente:

```
1 'root'@'linux'  
2 'root'@'localhost'  
3 'julio'@'localhost'  
4 'pepito'@'localhost'  
5 'julio3'@'localhost'  
6 'invento'@'localhost'
```

Figura A.10 Archivo para la evaluación 4

e. Evaluación 5

La evaluación 5 es la única en la que no es necesario añadir ningún dato más para realizarlo, ya que, sólo requiere el archivo de log que fue introducido en la pestaña de configuración. En este caso se evalúa la cantidad de accesos erróneos que se han producido en el sistema.

Por tanto, en el momento que se pulsa el botón de la evaluación esta se pone en marcha, como se puede ver en la siguiente imagen, en la cual, también se puede ver que el resto de evaluaciones se han ido realizando mientras se realizaba este manual.

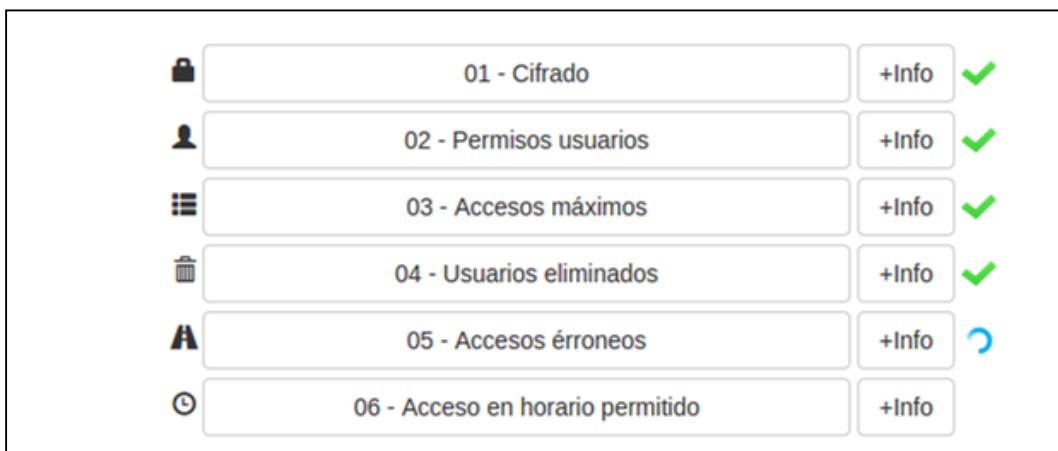


Figura A.11 Evaluación 5 en proceso

f. Evaluación 6

Por último, en la evaluación 6 si los usuarios han accedido al sistema durante su horarios laboral, para lo cual, será necesario añadir un archivo con los usuarios y su horario.

Figura A.12 Evaluación 6

Dicho archivo sigue el siguiente formato:

<nombre_de_usuario> in: <hora_entrada> // out: <hora_salida>

Y en este caso, se ha usado el siguiente archivo:

```

usuarios.txt
1 julio@localhost in: 16:00 || out: 21:00
2 root@localhost in: 11:30 || out: 12:12
3 julio2@localhost in: 17:00 || out: 17:51
4 julio3@localhost in: 12:00 || out: 14:45
5

```

Figura A.13 Archivo para la evaluación 6

Una vez realizadas las seis evaluaciones, se puede observar que todas tienen el tick verde comentado con anterioridad. El siguiente paso es ver la pestaña de informe final en la que se podrán ver y descargar los resultados.

01 - Cifrado	+Info	✓
02 - Permisos usuarios	+Info	✓
03 - Accesos máximos	+Info	✓
04 - Usuarios eliminados	+Info	✓
05 - Accesos erróneos	+Info	✓
06 - Acceso en horario permitido	+Info	✓

Figura A.14 Evaluaciones realizadas

3. Informe final.

En la pestaña del informe final se muestran los resultados obtenidos tras realizar las evaluaciones, esto se hace mediante gráficos de sectores y de barras. También se muestra un pequeño consejo sobre cómo interpretar los resultados obtenidos.

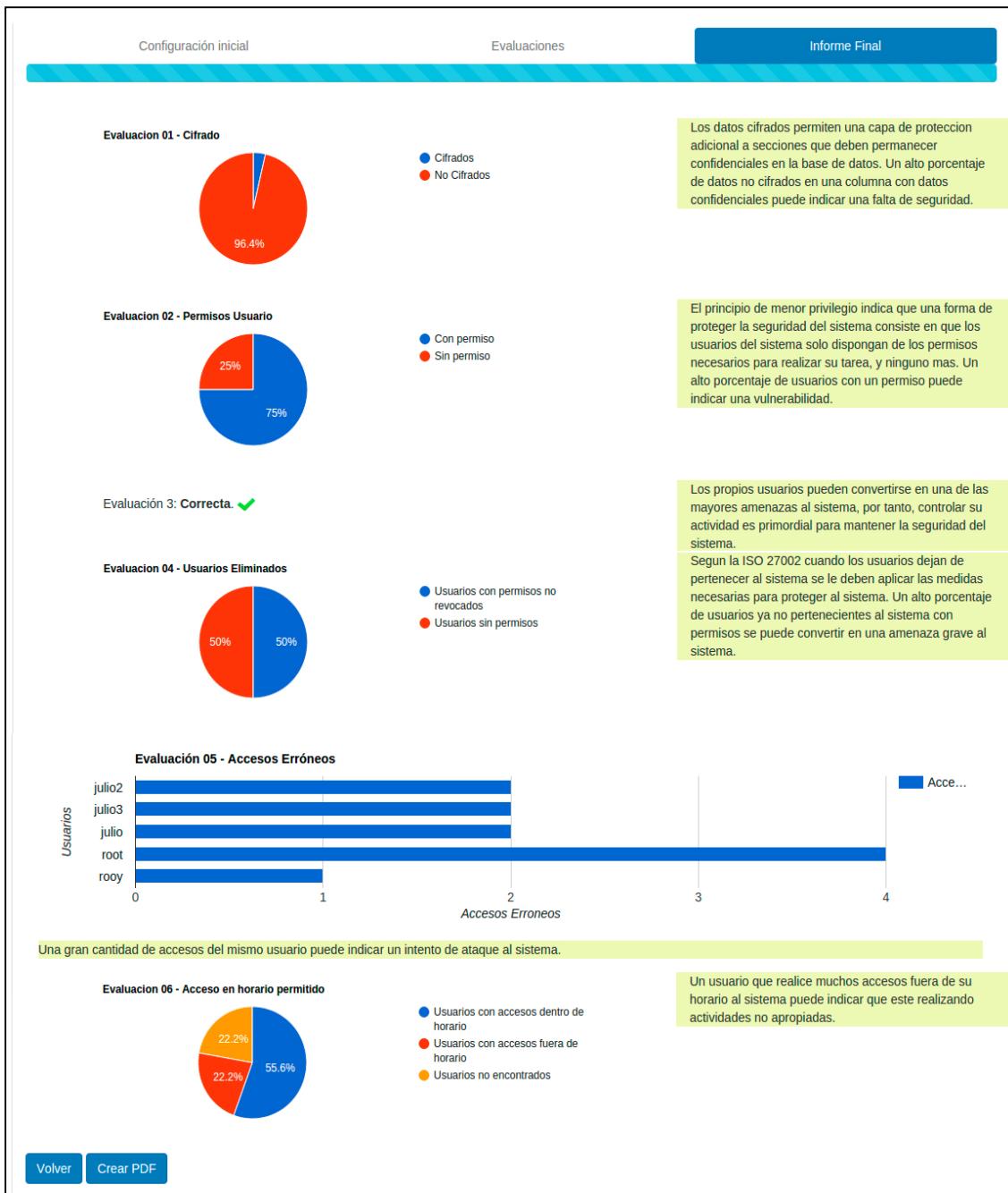


Figura A.15 Informe final

Como se puede ver, también existe la posibilidad de pasar los resultados a PDF para que el auditor los conserve en local. El PDF resultado se puede observar en la siguiente imagen.

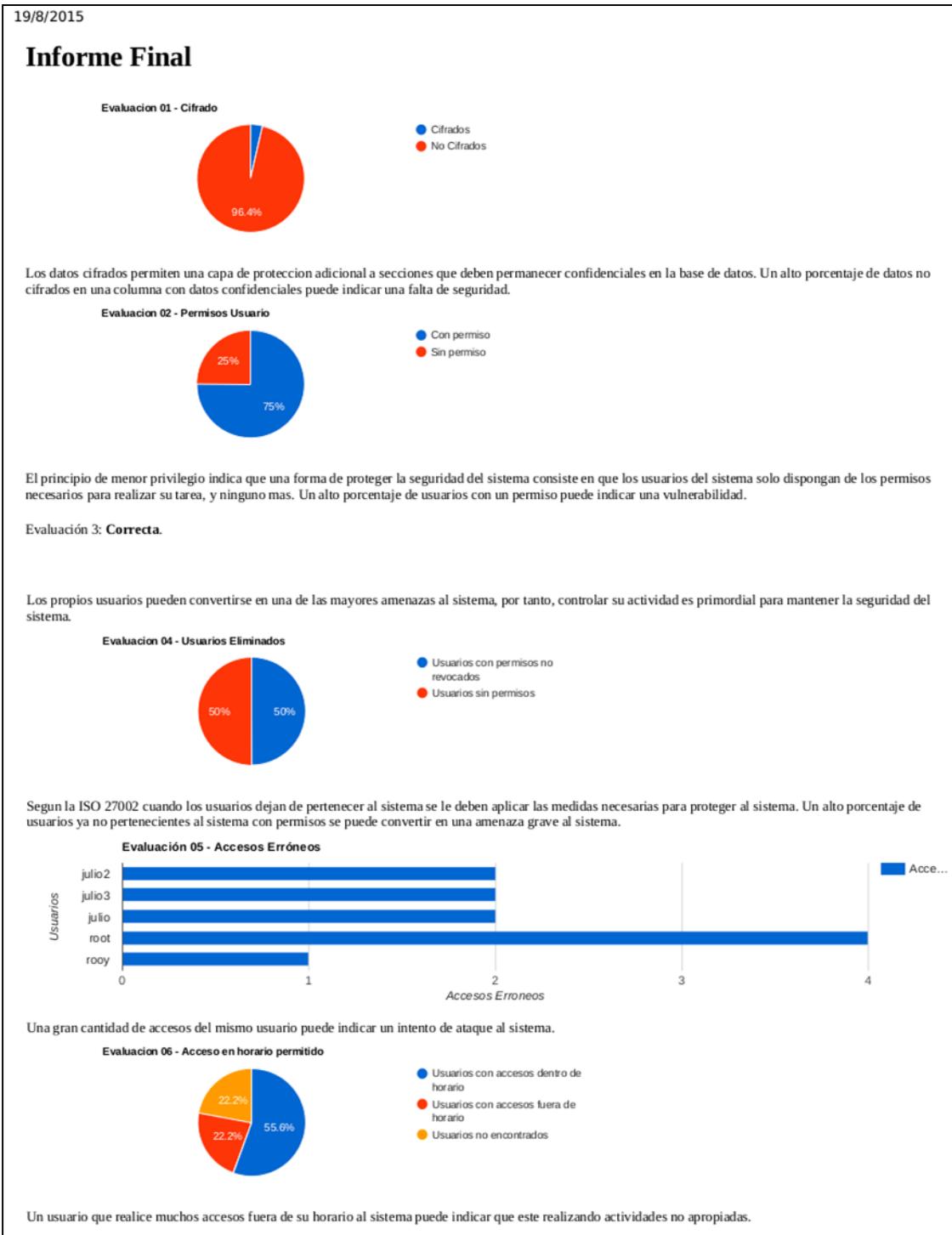


Figura A.0.16 Informe final en PDF

Anexo B. Manual de instalación de la herramienta

En este anexo se explicará cómo instalar los distintos programas necesarios para la ejecución de la herramienta. Para ello, primero será necesario realizar la instalación del entorno de programación NodeJS, seguir con el gestor de paquetes npm y finalmente el paquete Angular.

1. Instalación NodeJS y npm

Para la instalación de NodeJS y npm se ha seguido el manual indicado en [76], para ello, los pasos son los siguientes:

1. Añadir el repositorio necesario para instalar NodeJS.

```
sudo apt-get install python-software-properties  
sudo apt-add-repository ppa:chris-lea/node.js  
sudo apt-get update
```

2. Instalar el entorno NodeJS y el gestor de paquetes npm, mediante el siguiente comando, se instalarán ambos.

```
sudo apt-get install nodejs
```

3. Para comprobar que la instalación se ha realizado correctamente ejecutamos los siguientes comandos que nos devuelven el número de versión instalado.

```
node -v  
npm -v
```

2. Instalación Angular

Una vez tenemos instalado npm, para instalar Angular sólo será necesario ejecutar el siguiente comando:

```
npm install angular
```

Con angular instalado sólo será necesario añadir una línea para indicar la ruta del script de angular en el index.html.

3. Ejecución de la herramienta

Finalmente, para lanzar la herramienta simplemente será necesario ejecutar por línea de comandos el entorno de NodeJS pasándole como parámetro el archivo server.js.

```
npm install angular
```

Anexo C. Acrónimos

TFG	Trabajo de Fin de Grado
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
COBIT	Control Objectives for Information and related Technology
MAGERIT	Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información
SGBD	Sistema Gestor de Bases de Datos
HTML	HyperText Markup Languaje
CSS	Cascading Style Sheets
ZB	ZettaBytes
TI	Tecnologías de la Información
HDFS	Hadoop Distributed File System

