UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

TECNOLOGÍA ESPECÍFICA DE INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE GRADO

Sherdroplet Holmes:
Sistema Big Data para la detección de fraude y fugas en contadores de agua.

Francisco Parreño Heredia

Septiembre, 2016

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

Sherdroplet Holmes: Sistema Big Data para la detección de fraude y fugas en contadores de agua.

Autor: Francisco Parreño Heredia.

Director avanttic: Enrique Brandariz Reboredo.

Director UCLM: Ismael Caballero Muñoz-Reja.

Septiembre, 2016

PÁGINA DE CALIFICACIÓN

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE	VOCAL	SECRETARIO
Fdo.:	Fdo.:	Fdo.:

Resumen

En la actualidad, muchas empresas suministradoras y distribuidoras de agua almacenan las lecturas de sus contadores con el propósito de ser analizadas posteriormente, para obtener información de valor acerca de sus clientes (consumo diario, consumo por horas…). Pero la información que más interesa y por la que gastan un esfuerzo mayor es la detección de fugas y fraudes en sus contadores, ya que estas situaciones hacen que las empresas puedan perder mucho dinero.

Este proyecto se encarga de recoger los datos de lecturas de contadores durante un periodo de tiempo y analizarlas para encontrar posibles fraudes o fugas en dichos contadores. Estos análisis con tecnología transaccionales no se pueden abordar puesto que los análisis tardarían demasiado, por eso se ha optado por una solución Big Data.

Abstract

Currently, many suppliers and distributors of water store readings of their meters in order to be subsequently analyzed to obtain valuable information about their customers (daily consumption, consumption per hour ...). But the most interesting information should be used for the detection of leaks and fraud, since these situations cause companies to lose a lot of money.

This project is aimed to collecting data from meter readings over a period of time and enable various type of analysis to identify possible fraud or leaks in these meters. These transactional analysis technologies cannot be faced up with classical technologies due to the fact that the analysis would take too long, so Big Data solutions are instead deployed.

A mis padres y mi familia.

Por todo el apoyo inmenso que me habéis dado.

A Elena.

Por descubrirme que la felicidad tiene nombre.

AGRADECIMIENTOS

Me gustaría empezar por agradecer a mis padres todo, absolutamente todo lo que han hecho por mí, ya que de no ser por ellos no estaría escribiendo esto ahora mismo.

A mi hermano Álvaro que además de correr la misma sangre por nuestras venas, tengo el privilegio de tenerlo como amigo y confidente, gracias por estar apoyándome en los días malos.

A Ismael Caballero por ser además de mi tutor académico en este proyecto, un "hermano mayor" dándome consejos y tranquilizándome en momentos duros. Se ha ganado mi cariño a pulso.

A mis compañeros de FORTE con los que comencé esta aventura, Ana y Mario, que han sido un apoyo constante ya que los tres sabíamos las palabras que utilizar para darnos ánimos y seguir avanzando.

A avanttic Consultoría Tecnológica SL, a todas las personas que forman esta gran familia con la que me llevo vivencias personales y profesionales extraordinarias. Mención especial a Raúl que ha estado animándome de manera incondicional y he descubierto en él a un amigo.

Y para finalizar, a Elena por su apoyo no solo en este proyecto, sino en todos los momentos en los que he tenido días bajos y ha estado ahí para hacer de ese día otra razón más que me demuestra el por qué estoy compartiendo mi vida con ella.

Índice general

Índice de Figuras

Índice de Tablas

Índice de Listados

CAPÍTULO 1

INTRODUCCIÓN

Desde la antigüedad, el tratamiento del agua (suministro, potabilización,..) ha sido un problema para el hombre y más sabiendo de la importancia de ésta cuando es un bien esencial para la vida y, su acceso, un derecho humano reconocido por la Comunidad Internacional [1]. Del mismo modo, se puede ver como un recurso básico de la economía productiva de todos los países. Sin embargo, su escasez en ciertos territorios y circunstancias, hace que surja la necesidad de tener que realizar una gestión eficiente de los cauces y reservas, entre otras razones, para optimizar su consumo en diferentes dominios (consumo humano, riego, limpieza…). Esta gestión no está exenta de grandes desafíos, a los que las empresas suministradoras y distribuidoras tienen que responder con soluciones sostenibles, inteligentes, innovadoras y comprometidas tanto con el medio ambiente como con las necesidades básicas de los usuarios.

Una de las grandes amenazas de las empresas para realizar con éxito este tipo de gestiones son las fugas en el suministro y los fraudes en el consumo. Por eso, se hace necesario proteger las inversiones de las empresas que gestionan los recursos hidrológicos dotándoles de mecanismos que les ayuden a identificar, prevenir o paliar estas fugas y fraudes.

En el pasado, e incluso en la actualidad, el descubrimiento de esas fugas y fraudes se demora demasiado, lo que supone una pérdida de clientes y de capital muy grande para las empresas.

En este sentido, la comunidad científica ha proporcionado algunos algoritmos interesantes para la detección de estas fugas y fraudes [2]. No obstante, la implementación de esos algoritmos con las tecnologías existentes (e.g. tecnologías relacionales) no tiene un rendimiento aceptable, debido a múltiples factores como la velocidad en la que se generan los datos, el ruido que contienen y el volumen de los mismos.

Y aquí es donde las tecnologías Big Data empiezan a tener un papel más que relevante. Gartner define Big Data como *"gran volumen, gran velocidad y gran variedad de activos de información que exigen mecanismos innovadores y rentables de procesamiento de la información para mejorar la comprensión y la toma de decisiones* [3]*"*.

Este Trabajo Fin de Grado (TFG), realizado en el ámbito de un convenio FORTE con la empresa ***avanttic Consultoría Tecnológica, S.L.***, aborda el problema de desarrollar un entorno basado en tecnologías Big Data para la detección de fraude en consumo de agua. Este entorno está siendo desarrollado para una empresa proveedora de una ciudad española. El TFG se plantea no, como la solución final propiamente dicha, sino como una prueba de concepto en el que se involucran las tecnologías Oracle -de la que *avanttic* es Partner Platinum- que se usarán en el proyecto.

1.1. Contexto del problema.

Este Trabajo de Fin de Grado (TFG) se llevó a cabo en el contexto del programa **profESIonalízate (**http://webpub.esi.uclm.es/spa/paginas/empresas-profesionalizate**)** lanzado por la Escuela Superior de Informática (ESI) de Ciudad Real. Dicho programa busca el fortalecimiento de las competencias profesionales de los egresados de la ESI.

Se estructuró mediante la realización de prácticas en proyectos reales junto con el desarrollo del Trabajo Fin de Grado [4].

Las prácticas se han realizado en una de las sedes de la empresa ***avanttic Consultoría Tecnológica S.L.***, ubicada en Madrid, C\Capitán Haya 38, 6ºB, Edificio Cuzco II.

Por esta razón, muchas de las herramientas utilizadas en este proyecto involucran las tecnologías Oracle -de la que *avanttic* es Partner Platinum-. Esto servirá para que el alumno adquiera destrezas y conocimientos metodológicos y tecnológicos en la gestión e implantación de este tipo de proyectos, tanto en lo que se refiere al desarrollo de las herramientas software como al despliegue de las tecnologías necesarias.

1.2. Estructura del documento.

El documento de este Trabajo Fin de Grado (TFG de aquí en adelante) se compone de 7 capítulos y tres anexos.

- En el **Capítulo 1: Introducción** se describe cual es el problema abordado en el TFG y se presenta la solución propuesta. También se detalla el contexto y la estructura del documento completo.

- En el **Capítulo 2: Objetivos** se describe el objetivo principal del TFG, así como los objetivos parciales o sub-objetivos en los que se podrán desgranar hasta llegar a la solución final.

- En el **Capítulo 3: Estado del arte** se describe todas las tecnologías, técnicas y herramientas que se han aprendido a lo largo del TFG para la realización del mismo. Debido al marco tecnológico en el que se encuentra este proyecto se hablará de tecnologías Oracle debido a que *avanttic* es Partner Platinium, además de todo el ecosistema de Big Data.

- En el **Capítulo 4: Metodología de trabajo** se describe la metodología elegida para la realización del TFG, se discute el por qué se ha elegido esa metodología, se explican los elementos que la componen y como se aplica esa metodología para alcanzar los objetivos de este TFG.

- En el **Capítulo 5: Resultados** se muestran los resultados obtenidos de cada etapa al aplicar la metodología de trabajo citada en el Capítulo 4.

- En el **Capítulo 6: Conclusiones y Propuestas** se muestra la consecución de los objetivos propuestos en el Capítulo 2 una vez terminado el TFG. Asimismo, se presenta una conclusión del trabajo realizado, las posibles propuestas futuras y por último, una opinión personal de la realización de este TFG.

- En el **Capítulo 7: Bibliografía** se muestra la recopilación de toda la bibliografía utilizada en este TFG.

- Anexo A: Manual de usuario.
- Anexo B: Script generador de lecturas.
- Anexo C: Script para la clonación del terminal.

CAPÍTULO 2

OBJETIVOS DEL TFG

En este capítulo se describen el objetivo principal así como los objetivos parciales planteados en este TFG.

2.1. Objetivo principal.

El objetivo principal de este TFG es la realización de un entorno Big Data en el que se realicen análisis de detección de fraudes y fugas en contadores de agua y mostrar los resultados mediante una aplicación web, para que ayuden a la eficiencia de la organización.

2.2. Objetivos parciales

Para alcanzar el objetivo principal, se dividirá el objetivo principal en los siguientes objetivos parciales mostrados en la Tabla 1.

Objetivos Parciales	
1	Creación de un script que genere la simulación de las lecturas de los contadores digitales y posterior almacenamiento de datos.
2	Implementación de los algoritmos de detección de fraude usando el lenguaje estadístico R
3	Creación de una aplicación web que permita invocar y ejecutar los algoritmos de fraude ya implementados e interpretarlos mediante visualización de gráficas y tablas.
4	Generar un informe final en pdf para interpretar los resultados de los análisis.
5	Configuración y creación de un catálogo datos en la herramienta Big Data Discovery, proveniente de tablas Hive mediante la aplicación web.

Tabla 1. *Objetivos parciales del Proyecto.*

CAPÍTULO 3

ESTADO DEL ARTE

En este capítulo se describe la información necesaria que se ha estudiado e investigado para la realización de este TFG. Los puntos clave de los que va a tratar este capítulo son los siguientes:

- Contadores de agua.
- El ecosistema de Big Data.
- Estrategias para la detección del fraude en los contadores de agua.

3.1. Contadores de agua.

Un contador de agua [5] es un artefacto que permite contabilizar la cantidad de agua que pasa por él y es usado en las instalaciones residenciales e industriales de los acueductos para realizar la facturación a los usuarios correspondientes.

El contador suele ser propiedad de la entidad que suministra el agua lo cual tiene que verificar y asegurar que las mediciones son correctas, además de mantener el contador en perfecto estado con revisiones periódicas. La instalación de los contadores suele ser a través de toda la red, ya que esto permite un control más preciso a la hora de realizar un seguimiento de los consumos.

3.1.1. Tipos de contadores de agua.

Existen dos grandes grupos de contadores:

- **Contadores de lectura manual**: Son los contadores fijos convencionales. Esto dificulta a la hora de facturar un importe al cliente ya que son consumos estimados y suele ser habitual que se produzcan errores en esas facturas. La recogida de lecturas en esto contadores es de la manera tradicional, ya que el operario de la entidad suministradora debe personarse en el contador para registrar el consumo.

- **Contadores con lectura de medición automática:** Reciben la información de la lectura de forma electrónica. Facilitan los consumos reales de agua sin necesidad de que el operario tenga que visitar el contador. Esto hace que con las técnicas adecuadas se pueda sacar un valor de negocio a esas lecturas.

3.2. Ecosistema Big Data.

3.2.1. Concepto de Big Data.

El concepto de Big Data nace de la necesidad del uso de herramientas y técnicas innovadoras para el procesado de una gran cantidad de variedad y volumen de datos una velocidad tan alta, que los sistemas transaccionales no pueden afrontar [6]. Ya no vale únicamente con almacenar esos datos en almacenes de datos, sino que se quiere sacar todo el valor posible a esos datos, de ahí que nazca este nuevo concepto.

Una de las definiciones más usadas del término Big Data es la expuesta en el Capítulo 1, según Gartner Big Data se define como datos que tienen *"gran volumen, gran velocidad y/o gran variedad de conjuntos de datos que requieren nuevas formas de procesamiento para permitir una toma de decisiones mejorada, percepción del descubrimiento y optimización de procesos"* [7]. A esto se le denomina como las 3 V's del Big Data.

No obstante, esta definición se queda algo corta debido a que en la actualidad no solo consideramos Big Data como gran volumen, gran velocidad y gran variedad de datos, si no que se incorporan dos características más a esas 3 V's que son Valor y Veracidad ya que de nada vale tener mucha información almacenada si esta no es de calidad y no aporta valor una vez tratados los datos.

En Figura 1 la se puede ver en profundidad lo que engloba el concepto de Big Data.

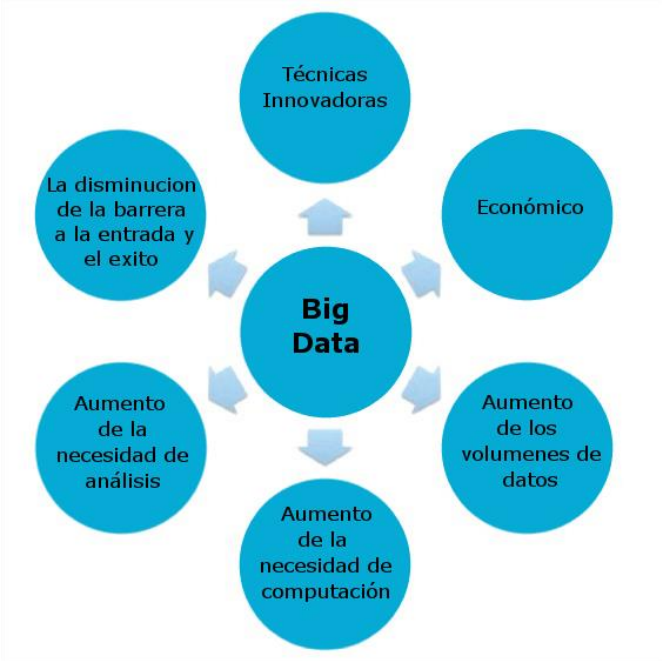

Figura 1. *Características Big Data. Adaptada de* [8].

3.2.2. Paradigma de programación MapReduce.

MapReduce [9] es un paradigma de programación que permite a los desarrolladores el procesamiento de grandes cantidades de datos de forma paralela y distribuida. Para ello consta de dos funciones principales:

- La función Map que recibe como parámetros un par (clave, valor) y devuelve una lista de pares.
- La función Reduce se aplica en paralelo para cada grupo creado por la función Map. Esta función se llama una vez para cada clave única de la salida de la función Map. Junto con esta clave, se pasa una lista de todos los valores asociados con la clave para que pueda realizar alguna fusión para producir un conjunto más pequeño de los valores.

La Figura 2 muestra el esquema.

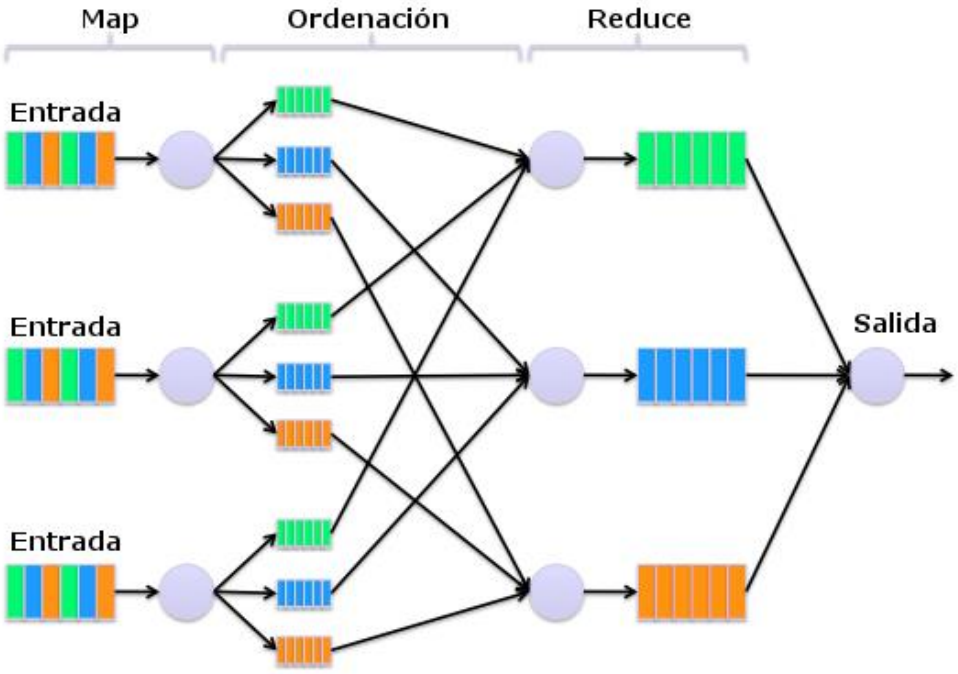

Figura 2. *Esquema del funcionamiento de MapReduce.*

3.2.2.1. Apache Hadoop.

Apache Hadoop [10] es un framework que permite realizar un procesamiento distribuido de grandes cantidades de datos a través de distintos *clusters*. Está diseñado para ser escalable desde un único servidor a miles de servidores, cada uno ofreciendo computación y almacenamiento local.

El modelo de programación para dar soporte a esta computación paralela sobre grandes colecciones de datos en grupos de computadoras se le denomina MapReduce ya mencionado anteriormente, que fue utilizado por Google.

Se dice que MapReduce es el corazón de Apache Hadoop por lo que ambos términos van ligados.

Apache Hadoop incluye un sistema distribuido de ficheros (HDFS), que divide los datos de entrada y los almacena en distintos nodos.

3.2.2.2. HDFS.

Hadoop Distributed File System (HDFS) [11] es un sistema de archivos distribuido diseñado para ejecutarse en hardware. Tiene muchas similitudes con los sistemas de archivos distribuidos existentes. Sin embargo, las diferencias con respecto a otros sistemas de archivos distribuidos son significativas. HDFS es altamente tolerante a fallos y está diseñado para ser implementado en hardware de bajo costo. HDFS proporciona un alto rendimiento de acceso a datos de la aplicación y es adecuado para aplicaciones que tienen grandes conjuntos de datos.

3.3. Estrategias para la detección del fraude.

Se dice que el consumo marcado por un contador es fraudulento, cuando se factura menos importe de lo que realmente se ha consumido y el cliente está involucrado en esa trampa.

Para evitar estos problemas se deben tomar medidas de detección que ayuden a detectar esos fraudes y erradicarlos de inmediato.

En este sentido, la comunidad científica ha proporcionado algunos algoritmos interesantes para la detección de estas fugas y fraudes marcados por estos índices [12]:

- Una caída progresiva en el consumo
- Una caída súbita en el consumo y normalización del mismo.
- Un consumo anual anormalmente bajo.

3.3.1. Una caída progresiva en el consumo.

Un síntoma evidente de una lectura anormal para un contador es una disminución en el consumo. Este tipo de disminución puede deberse a un decremento real en el consumo (por ejemplo, debido a un cambio en el número de personas que viven en un hogar, fallo del contador o la manipulación voluntaria del contador (fraude).

Para detectar la disminución del consumo debido al fraude, se desarrolló una solución basada en el coeficiente de correlación de Pearson, este coeficiente es una medida de la relación lineal entre dos variables aleatorias cuantitativas. Estas variables que se van a relacionar son el paso del tiempo y el consumo en cada lectura, es decir, a medida que pasa el tiempo si

existe una conducta fraudulenta la relación entre ambas variables debe acercarse al coeficiente de correlación de Pearson de valor -1.

3.3.2. Una caída súbita en el consumo y normalización del mismo.

La detección de aquellos clientes cuyo consumo se había estabilizado después de una caída inicial en el consumo también es de interés. Estos clientes no se pueden detectar con la solución propuesta anteriormente por lo que se ha buscado otra alternativa.

La alternativa es la siguiente:

- Primero se divide la muestra en cuatro trimestres:
    - Primer trimestre: Enero, febrero y marzo.
    - Segundo trimestre: Abril, mayo y junio.
    - Tercer trimestre: Julio, agosto y septiembre.
    - Cuarto trimestre: Octubre, noviembre y diciembre.
- Después se realiza la desviación típica cogiendo los datos de dos trimestres adyacentes:
    - Primer trimestre y Segundo trimestre.
    - Segundo trimestre y Tercer trimestre.
    - Tercer trimestre y Cuarto trimestre.
- Para finalizar se calcula el porcentaje de variación de consumo que han tenido que han tenido estos dos trimestres adyacentes.

3.3.3. Un consumo anual anormalmente bajo.

Según el Instituto Nacional de Estadística, el consumo medio de agua de los hogares se sitúa en unos 130 litros por habitante y día [13], lo que supone una media anual de unos 41 m^3 por habitante.

Aunque el consumo de agua puede ser muy desigual y depende de múltiples factores (hogares cerrados, el cambio de los negocios, los hábitos de gasto, aparatos utilizados, el uso de riego). Puesto que un bajo consumo no corresponde necesariamente a una manipulación del contador o algún tipo de fallo del equipo de medición, este número (130 litros de agua

por persona/día) se alcanzó después de un profundo estudio estadístico por lo que era un dato muy relevante para ser utilizado para la detección.

CAPÍTULO 4

METODOLOGÍA DE TRABAJO

En este capítulo se describe el método de trabajo utilizado para el desarrollo de este Trabajo Fin de Grado. La metodología que se ha aplicado para obtener el sistema **Sherdroplet Holmes** es Scrum. Es una metodología ágil de gestión de proyectos que se basa en un modelo iterativo e incremental. Se ha escogido esta metodología por ser la que más se adapta a la metodología seguida por la empresa *avanttic Consultoría Tecnológica* en proyectos de índole semejante además de por su gran flexibilidad y su capacidad de adaptación a los cambios de las circunstancias relacionadas con el proyecto.

4.1. Scrum.

Scrum es una metodología ágil que proporciona un marco de trabajo para la gestión de proyectos [14], no se basa en el seguimiento de un plan sino en la adaptación continua a las circunstancias de la evolución del proyecto. La captura de requisitos se basa en la creación de "*historias de usuario*" que son una manera simple de describir tareas, en una o dos frases. A todo el conjunto de historias de usuario se le llama pila de producto [15].

Estos requisitos o historias de usuarios de dividen en iteraciones o fases denominadas Sprints. A su vez, cada uno de estos Sprints está formado por una lista de tareas necesarias para el cumplimiento de los requisitos.

4.1.1. El equipo en Scrum.

En el equipo Scrum se definen tres roles perfectamente acotados: Propietario del Producto, Scrum Master y Equipo de desarrollo. Estos equipos son auto-organizados y multifuncionales. Esto está diseñado así para optimizar la flexibilidad, la creatividad y la productividad. Además de estos tres roles, se puede hablar de un cuarto rol externo al desarrollo del producto que serían los Interesados o Stakeholders.

4.1.1.1. Propietario del Producto.

El propietario del producto representa a todos los interesados en el producto. Es el responsable de lograr el mayor valor de producto para los clientes, usuarios y resto de implicados. Sus responsabilidades más destacadas son:

- Responsable de la visión de producto, es decir, el que entiende el valor de negocio de las características del sistema y la gestión económica de su desarrollo.
- Es el nexo de conexión entre el equipo y los interesados.
- Decide qué historias de usuario se debe incluir en la pila de producto, priorizarlas y de finalmente validarlas.
- Optimizar el valor que el Equipo de Desarrollo realiza.

El propietario del producto es el director del proyectando en avanttic Consultoría Tecnológica.

4.1.1.2. Scrum Master.

El Scrum Master es el encargado de asegurar que Scrum se entienda y se lleve a cabo. El Scrum Master no adopta la figura de jefe, sino que actúa de líder el cual está al servicio del equipo Scrum además de ayudar a las personas externas al desarrollo del proyecto a entender que iteraciones pueden ser útiles y cuáles no.

El Scrum Master es el director del proyectando en la UCLM.

4.1.1.3. Equipo de desarrollo.

El equipo está compuesto por los desarrolladores, que convertirán las necesidades del propietario del producto en un conjunto de nuevas funcionalidades, modificaciones, o incrementos del producto software final. Solo los miembros del equipo de desarrollo participan en la creación del incremento. Se tratan de grupos auto-organizativos y no son dirigidos por personas ajenas al equipo.

El equipo de desarrollo está formado por el proyectando.

4.1.1.4. Interesados.

Son personas que están implicadas con el proyecto, pero son ajenas o no están comprometidas con el desarrollo del mismo. Estos interesados son los que hacen posible la consecución del proyecto y para los quienes el proyecto producirá el beneficio acordado que justifica su desarrollo. Sólo participan directamente en las revisiones de los sprints.

4.1.2. Artefactos de Scrum.

4.1.2.1. Sprint.

Como se ha mencionado anteriormente, a cada iteración o fase se le denomina Sprint. Su duración varía entre 1 y 4 semanas dependiendo de lo decidido por el Scrum Master.

Durante cada Sprint, se desarrolla una parte del producto y al final de cada Sprint se obtiene un incremento del producto final que puede ser puesto en producción.

La Tabla 2 muestra los tipos de reuniones dentro de la metodología Scrum.

Tipo de Reunión	Participantes	Tiempo
Planificación Sprint	Todos los roles.	< 8h
Diarias	Equipo de desarrollo y Scrum Master.	< 15’
Revisión del Sprint	Propietario del Producto y Scrum Master.	< 4h
Retrospectiva del Sprint	Equipo de desarrollo y Scrum Master.	< 4h

Tabla 2. Tipos de reuniones en Scrum.

4.1.2.2. Historias de Usuario.

Las historias de usuario son, principalmente, lo que el cliente o el usuario quiere que se implemente, es decir, son una descripción breve de una funcionalidad software tal y como la percibe el usuario. Son independientes unas de otras.

4.1.2.3. Pila de producto.

Es una lista de *historias de usuario,* que se incorporarán al producto software a partir de incrementos sucesivos. Es decir, sería similar a un listado de requisitos de usuario que representa lo que el cliente espera.

4.1.2.4. Pila de Sprint.

Es la lista de trabajos o tareas derivadas de las historias de usuario, que debe realizar el equipo durante el sprint.

La Figura 3 muestra un esquema de la metodología Scrum con los artefactos más importantes.

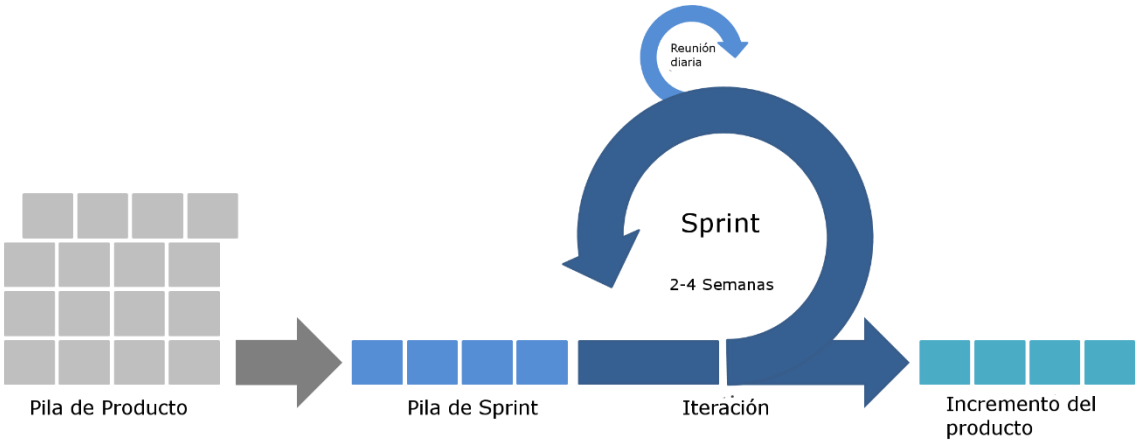

Figura 3. Esquema metodología Scrum.

4.1.2.5. Gráfico Burn-Up.

Este gráfico proyecta en el tiempo la construcción del producto, una línea marca la estimación inicial que se hizo del sprint, mientras que una segunda línea marca el trabajo que se ha estado realizando durante el sprint. Este gráfico se mostrará al final de cada sprint para evidenciar si las estimaciones iniciales han sido acordes con lo posteriormente realizado.

4.2. Aplicación de Scrum.

Como se ha dicho en el apartado 4.1, no se basa en el seguimiento de un plan sino en la adaptación continua a las circunstancias de la evolución del proyecto lo que lleva a que sea mucho más fácil solucionar errores a tiempo para no arrastrarlos en las siguientes iteraciones. Además, no es necesario esperar a la finalización del ciclo de vida para obtener un incremento o una versión apta para ser probada por el cliente. Con esto se consigue que el

cliente esté más involucrado y contento con los pasos que va dando el proyecto hacia el producto final.

Pero como ya es sabido, cada metodología hay que adaptarla a las necesidades del proyecto en concreto. avanttic, que, para recordarlo, es la empresa en la que se está realizando este TFG, tiene un amplio conocimiento en la elaboración de este tipo de proyectos por lo que con una breve reunión entre los responsables del proyecto se decidió que la utilización de esta metodología se ajustaba mejor a las necesidades de dicho proyecto.

El equipo Scrum implicado en este proyecto está formado por tres personas, Ismael Caballero Muñoz-Reja como Scrum Master, Enrique Brandariz Reboredo como Propietario del Producto y Francisco Parreño Heredia como equipo de desarrollo. Se ha explicado que el equipo de desarrollo lo forman más de una persona, por lo que aquí viene la primera adaptación significativa, puesto que ahora el equipo de desarrollo lo forma una única persona. Esto tiene como consecuencia la eliminación de reuniones diarias, ya que no tienen sentido alguno mantenerlas. Por tanto, los sprints se han re-estructurado con una planificación inicial del sprint, y dos reuniones finales, la revisión del sprint y la retrospectiva del sprint.

Usando como técnica de estimación Planning Poker basada en el consenso (con la variante de la sucesión de Fibonacci) y teniendo en cuenta el correspondiente valor para la organización marcada por el propietario del producto, se han observado un conjunto de historias de usuario que conforman la pila de producto:

- HdU1: *"Yo como usuario quiero poder registrarme e identificarme en la aplicación según mi rol que desempeñe en la organización"*. (Valor de negocio 85; Estimación del esfuerzo: 144).
- HdU2: *"Yo como analista de fraude quiero poder crear desde la aplicación un catálogo de datos en la herramienta Big Data Discovery abstrayéndome de toda la lógica que eso implica."*. (Valor de negocio 75; Estimación del esfuerzo: 21).
- HdU3: *"Yo como analista de fraude quiero poder identificar los posibles contadores sospechosos de fraude y posibles fugas".* (Valor de negocio 95; Estimación del esfuerzo: 89).

- HdU4: *"Yo como operario quiero obtener en la aplicación la dirección de los contadores de agua que son sospechosos de fraudes o fugas para poder personarme en la ubicación de dicho contador".* (Valor de negocio 90; Estimación del esfuerzo: 34).
- HdU5: *"Yo operario quiero poder obtener un informe en formato pdf de los resultados de los análisis".* (Valor de negocio 80; Estimación del esfuerzo: 13).

Una vez obtenidas las historias de usuario y su correspondiente valor para la organización, se han planificado cada una de ellas en los distintos sprints que van a formar parte del total desarrollo del proyecto como se muestra en la Tabla 3.

Sprint	Historia de usuario (HdU)	Estimación del esfuerzo (horas)	Valor de negocio
0	Planificación inicial	44,5	-
1	HdU1	144	85
2	HdU3	89	95
3	HdU4	34	90
4	HdU5	13	80
5	HdU2	21	75
6	Finalización	89	-

Tabla 3. Sprints.

4.3. Marco tecnológico para el desarrollo del proyecto.

En esta sección se exponen las tecnologías, herramientas y frameworks necesarios para el desarrollo de este TFG, además se hará una breve explicación de cada una de ellas indicando cual ha sido la finalidad de cada una de ellas y se añadirá la versión utilizada de cada una de ellas si fuese necesario.

4.3.1. Herramientas para la gestión de proyectos.

4.3.1.1. Bitbucket.

Es el servicio de alojamiento web, en el cual está alojado este TFG. Bitbucket ofrece tanto repositorios públicos como privados, en los que permite la gestión de código fuente, el trabajo en grupo y la integración con herramientas de gestión de proyectos. Este TFG se

aloja en un repositorio privado al cual se puede dar permisos para que sea visible por las demás partes del proyecto.

El repositorio donde se encuentra el código de Sherdroplet Holmes es:

https://Packandalv@bitbucket.org/Packandalv/sherdropletholmes_app.git

4.3.1.2. Git

Es un sistema de control de versiones diseñado para el manejo y mantenimiento de versiones de aplicaciones de una forma rápida y eficiente. En este TFG, Git se integra perfectamente con el repositorio de Bitbucket y con el entorno de desarrollo integrado de R llamado RStudio, ya que tiene la opción de utilizar Git a la hora de crear un nuevo proyecto.

4.3.1.3. EGI (hErramienta de Gestión Interna de avanttic).

Es una herramienta interna de la empresa ***avanttic Consultoría Tecnológica***, en la que se está realizando este TFG. Esta herramienta es utilizada en este proyecto para imputar el número de horas realizadas día a día y a que tarea corresponde, esto actuará de historial para más adelante poder calcular las horas realmente trabajadas y poder contrastar esa información con la que se estimó en un principio antes de iniciar el TFG. Esto llevará a la realización de gráficos burn-up mencionados en el apartado 4.2.1 que muestran claramente el trabajo realizado con el estimado.

4.3.1.4. Microsoft Outlook 2013.

Microsoft Outlook [16] es una aplicación de gestión de correo, así como agenda personal, que permite comunicar al equipo de desarrollo con cuantas personas se quiera a través de correos electrónicos.

Debido a que el autor de este TFG es alumno de la Universidad de Castilla-La Mancha tiene acceso a todos los componentes del paquete Office 365. Esta herramienta se ha utilizado para establecer reuniones de seguimiento del proyecto.

4.3.1.5. Microsoft Excel 2013.

Microsoft Excel [17] es una aplicación informática desarrollada y distribuida por Microsoft Corp. Se trata de un software que permite trabajar con datos numéricos, es decir, se puede realizar tareas contables y financieras gracias a sus funciones, desarrolladas específicamente para ayudar a crear y trabajar con hojas de cálculo.

En el desarrollo de este TFG, Microsoft Excel se ha utilizado para la creación de los gráficos Burn up en la finalización de cada sprint, para mostrar de una manera visual la comparativa entre el trabajo realizado y el trabajo estimado

4.3.1.6. Hatjitsu.

Hatjitsu [18] es una aplicación web que permite realizar las estimaciones mediante *Planning Poker* [19] entre los distintos miembros del equipo Scrum. Utiliza la metáfora de habitaciones para que así cada reunión sea privada para cada equipo Scrum.

En este TFG se ha utilizado para la estimación de los distintos sprints que conforman el proyecto.

4.3.2. Herramientas para el modelado de software.

4.3.2.1. Visual Paradigm.

Visual Paradigm [20] 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 clase y diagramas de componentes.

4.3.2.2. Balsamiq Mockups.

Balsamiq Mockups [21] es una aplicación utilizada para la creación del diseño de bocetos y borradores. Su mayor característica es la de permitir con sencillez y rapidez crear un primer

esquema de cómo será tu aplicación. Además, se pueden exportar todos los bocetos en pdf y png, entre muchas opciones más.

Esta aplicación será utilizada en aquellos sprints en los que sea necesario mostrar cual va siendo la interfaz que finalmente va a tener nuestra aplicación web.

4.3.3. Herramientas y tecnologías para el desarrollo del proyecto.

4.3.3.1. Apache Hadoop.

Como ya se ha explicado en el Capítulo 3, Apache Hadoop

En este TFG Apache Hadoop va a ir cohesionado con un paquete de R llamado Oracle R Connector for Hadoop (ORCH) que utilizando llamadas de ese paquete va a comunicarse con Hadoop y con el HDFS para realizar los análisis de detección y almacenar los resultados.

4.3.3.2. Python.

Python [23]es un lenguaje de programación interpretado cuya filosofía hace es hacer una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Python será usado en este TFG para la realización de uno o varios scripts que simulen los consumos de los contadores de una manera en que los datos puedan ser luego analizados.

4.3.3.3. R.

R [24] es un entorno y lenguaje de programación con un enfoque al análisis estadístico. Es un leguaje que está enfocado en múltiples campos tales como la minería de datos, la investigación biomédica y las matemáticas financieras. Esto contribuye la posibilidad de cargar diferentes bibliotecas y paquetes con funcionalidades de cálculo o visualización de gráficos.

En este TFG, R es la columna vertebral ya que es el lenguaje utilizado para toda la aplicación web (con sus correspondientes paquetes) y para los análisis de detección de fraude y fugas.

4.3.3.4. R Studio.

R Studio [25] es el entorno de desarrollo integrado (IDE) por excelencia para R. Funciona con la versión estándar de R disponible en CRAN [26]. Es una herramienta potente que soporta procedimientos y técnicas requeridas para realizar análisis de alta calidad.

Será usado en este TFG como entorno de desarrollo para la creación de este proyecto.

4.3.3.5. Shiny y Shinydashboard.

Shiny [27] es un paquete de R que sirve para crear fácilmente aplicaciones web interactivas que permiten a los usuarios interactuar con sus datos sin tener que manipular código.

Shiny se basa en la programación reactiva que vincula los valores de entrada con los de salida. Además, dispone de pequeños módulos pre-construidos haciendo la aplicación más atractiva.

Shinydashboard [28] es una variante de Shiny que hace que la aplicación tenga un aspecto visual de un *dashboard.*

Para este TFG se han utilizado ambos paquetes para la realización de la aplicación además del paquete Oracle R Connector for Hadoop (ORCH) para la realización de los análisis de detección.

4.3.4. Herramientas y tecnologías para la base de datos.

4.3.4.1. Oracle Database 12c.

Oracle Database 12c [29] es un sistema de gestión de base de datos objeto-relacional desarrollado por Oracle. Este sistema ofrece soporte para transacciones, estabilidad, escalabilidad y soporte multiplataforma entre otras.

Se ha utilizado para el almacenamiento de los usuarios que acceden en la aplicación y también para el almacenamiento de los datos de los inmuebles en los que se encuentra cada uno de los contadores.

4.3.4.2. SQLDeveloper.

Oracle SQL Developer [30] es una herramienta gráfica que mejora la productividad y simplifica las tareas de desarrollo para base de datos Oracle. Usando Oracle SQL Developer, se puede navegar, editar y crear objetos de base de datos Oracle, ejecutar sentencias SQL, editar y depurar PL/SQL, construcción de PL/SQL de pruebas unitarias, ejecutar informes y colocar archivos bajo control de versiones.

En este TFG se ha utilizado para tener una versión más visual de los datos almacenados en la base de datos.

4.3.5. Herramientas para la elaboración de la memoria.

4.3.5.1. Microsoft Word 2013.

Microsoft Word [31] es un procesador de texto desarrollado por Microsoft perteneciente a la suite de aplicaciones ofimáticas de Microsoft Office.

Esta herramienta ha sido utilizada para la realización de la memoria, ya que permite que la revisión por parte del tutor sea mucho más fácil además de poder anotar comentarios para correcciones posteriores.

4.3.5.2. Zotero.

Zotero [32] es un programa de software libre para la gestión de referencias bibliográficas. Zotero permite al usuario recolectar, administrar y citar investigaciones de todo tipo, para ello importa los datos recogidos en internet y los almacena en una biblioteca propia.

En este TFG se ha utilizado para la realización de la bibliografía, desde añadir las citas hasta crear el índice. Este programa se completa con un plugin que existe para Firefox y compatible con Microsoft Word.

4.3.5.3. Gimp 2.

GIMP (GNU Image Manipulation Program) [33] es una herramienta de edición de imágenes. Es software libre bajo licencias GNU GPL y GNU LGPL.

Se ha utilizado para la modificación y creación las imágenes usadas en este TFG.

4.3.5.4. PrtScr.

PrtScr [34] en una aplicación disponible solo para Windows que nos permite hacer capturas de pantalla de una forma rápida y sencilla.

En este TFG se ha utilizado para realizar las capturas de la interfaz de usuario para la creación del manual de usuario.

CAPÍTULO 5

RESULTADOS

En este capítulo se presentan los resultados obtenidos tras la ejecución del plan de proyecto desarrollado en el Capítulo 4, satisfaciendo así los objetivos descritos en el Capítulo 2.

5.1. Sprint 0.

El objetivo principal de este Sprint es la identificación de tareas contenidas en la pila de producto, la creación de un plan de proyecto y mostrar la arquitectura de la aplicación.

Además de añadir cuestiones como un plan de gestión de riesgos.

5.1.1. Equipo Scrum.

El equipo Scrum establecido desde el inicio es el siguiente:

- **Propietario del Producto**: El responsable de este rol es Enrique Brandariz Reboredo, de *avanttic*
- **Scrum Master:** El responsable del funcionamiento de la metodología Scrum es Ismael Caballero Muñoz-Reja, de la UCLM
- **Equipo de desarrollo:** El encargado de desarrollar el producto es Francisco Parreño Heredia.

5.1.2. Planificación del Sprint.

En este primer Sprint se realizan esta lista de tareas que se muestra en la Tabla 4:

Sprint	Lista de Tareas	Objs*	Horas estimadas
0	T1. Identificar los roles del sistema.	O.1	2,5
	T2. Listar historias de usuario.		10
	T3. Estimación de las historias de usuario.		3
	T4. Creación del documento Anteproyecto.		29,5
		Número total de horas:	**44,5**

Tabla 4. Planificación del Sprint 0.

5.1.3. Captura de los requisitos.

Para la elicitación de los requisitos del sistema Sherdroplet Holmes se han llevado a cabo varias entrevistas con el propietario del producto. Estas entrevistas han sido de vital importancia para saber cuál es el alcance del proyecto y para averiguar qué datos se deben guardar de las lecturas de los contadores de agua ya que no se parte de datos reales, sino que hay que simularlos.

5.1.4. Roles del sistema Sherdroplet Holmes.

El sistema Sherdroplet Holmes tiene dos roles de usuario, los cuales dejan muy bien delimitados a que puede acceder cada uno y a que pueden acceder ambos.

Por un lado, se tiene el rol de “Analista de fraude” el cual se encarga de la parte del tratado de los datos y la obtención de los resultados finales de la ejecución de los análisis de detección de fraude y fugas.

Por otro lado, tenemos el rol de “Operario” que se le notifica cuales han sido los contadores con mediciones anómalas para así personarse en el domicilio del cliente y tratar de resolver la incidencia.

La Tabla 5 muestra las historias de usuario que podrán realizar cada usuario.

Historia de usuario	Analista de fraude	Operario
HdU1	SI	SI
HdU2	SI	NO
HdU3	SI	NO
HdU4	NO	SI
HdU5	NO	SI

Tabla 5. Relación de los usuarios con las historias de usuario.

5.1.5. Pila del Producto.

A continuación, en las Tabla 6 a Tabla 10 se describen las historias de usuario previamente citadas:

Historia de Usuario	
Número: 1	**Usuario:** Analista de fraude y Operario
Nombre de la historia: Gestión de los roles de usuario	
Prioridad de negocio: 85	
Esfuerzo: 144	**Sprint asignado:** 1
Programador responsable: Francisco Parreño Heredia	
Descripción: Como usuario quiero poder registrarme e identificarme en la aplicación según mi rol que desempeñe en la organización.	
Precondición: No es necesaria ninguna precondición para la realización de la historia de usuario.	
Postcondición: Tras la realización de este esta historia de usuario, el usuario podrá identificarse con su correspondiente rol.	
Tareas (T): T1. Decidir el sistema gestor de base de datos óptimo para que la integración con los lenguajes de programación utilizados en el proyecto sea la mejor posible. T2. Modelado de la base de datos y las tablas para almacenar a los usuarios. T3. Implementación de la funcionalidad registrar usuario. T4. Implementación de la funcionalidad loguear usuario. T5. Implementación de la funcionalidad error al loguear.	
Artefacto o producto generado: • Funcionalidad login usuarios. • Funcionalidad login de usuarios erróneo. • Funcionalidad registrar usuarios.	

Tabla 6. *Historia de Usuario 1.*

Historia de Usuario	
Número: 2	**Usuario:** Analista de fraude
Nombre de la historia: Creación catálogo en Big Data Discovery.	
Prioridad de negocio: 75	
Esfuerzo: 21	**Sprint asignado:** 5
Programador responsable: Francisco Parreño Heredia	
Descripción: Como analista de fraude quiero poder crear desde la aplicación un catálogo de datos en la herramienta Big Data Discovery abstrayéndome de toda la lógica que eso implica.	
Precondición: Haber iniciado sesión en la aplicación teniendo el rol de analista de fraude y tener instalada la herramienta Big Data Discovery.	
Postcondición: El analista de fraude habrá creado el catálogo de datos dentro de Big Data Discovery.	
Tareas (T): T1. Implementación para la posterior creación de un proyecto en Big Data Discovery. T2. Integración del script en la aplicación.	
Artefacto o producto generado: • Funcionalidad creación del catálogo.	

Tabla 7. Historia de Usuario 2.

Historia de Usuario	
Número: 3	**Usuario:** Analista de fraude
Nombre de la historia: Identificar fugas y fraudes.	
Prioridad de negocio: 95	
Esfuerzo: 89	**Sprint asignado:** 2
Programador responsable: Francisco Parreño Heredia	
Descripción: Como analista de fraude quiero poder identificar los posibles contadores sospechosos de fraude y posibles fugas.	
Precondición: Haber iniciado sesión en la aplicación teniendo el rol de analista de fraude.	
Postcondición: El analista de fraude tendrá en la aplicación los resultados de cada análisis de detección de fraude y fugas.	
Tareas (T): T1. Decidir que lenguaje utilizar para el script que genere las lecturas de los contadores. T2. Estudio de los tipos de fraude que se pueden cometer. T3. Creación del script que simule la generación de lecturas de contadores de agua. T4. Implementación del script del primer análisis de fraude. T5. Implementación del script del segundo análisis de fraude. T6. Implementación del script del tercer análisis de fraude. T7. Integrar los scripts con la aplicación.	
Artefacto o producto generado: • Gráfica para la representación del primer análisis. • Tabla para la representación del segundo análisis. • Tabla para la representación del tercer análisis.	

Tabla 8. *Historia de Usuario 3.*

Historia de Usuario	
Número: 4	**Usuario:** Operario
Nombre de la historia: Visualización de los análisis	
Prioridad de negocio: 90	
Esfuerzo: 34	**Sprint asignado**: 3
Programador responsable: Francisco Parreño Heredia	
Descripción: Como operario quiero obtener en la aplicación la dirección de los contadores de agua que son sospechosos de fraudes o fugas para poder personarme en la ubicación de dicho contador.	
Precondición: • Haber iniciado sesión en la aplicación con el rol de operario. • Que el analista de fraude haya iniciado los scripts para el análisis de los datos de consumo.	
Postcondición: El operario obtiene únicamente los contadores sospechosos de fraude y fugas, y la dirección de los mismos.	
Tareas (T): T1. Implementación de la funcionalidad obtener contadores sospechosos.	
Artefacto o producto generado: • Funcionalidad obtención de los contadores y sus direcciones.	

Tabla 9. Historia de Usuario 4.

Historia de Usuario	
Número: 5	**Usuario:** Operario
Nombre de la historia: Creación documento pdf.	
Prioridad de negocio: 80	
Esfuerzo: 13	**Sprint asignado:** 4
Programador responsable: Francisco Parreño Heredia	
Descripción: Como analista de fraude y operario quiero poder obtener un informe en formato pdf de los resultados de los análisis.	
Precondición: • Haber iniciado sesión como analista de fraude u operario • Haber obtenido los resultados de los análisis de los consumos.	
Postcondición: Obtención de los documentos pdf según el rol del usuario.	
Tareas (T): T1. Generar varios pdf que tengan el contenido de los resultados de los análisis y de los contadores con sus respectivas direcciones.	
Artefacto o producto generado: • Funcionalidad para la obtención del documento pdf.	

Tabla 10. *Historia de Usuario 5.*

5.1.6. Plan de proyecto.

En este primer sprint se ha obtenido la planificación del proyecto con todas las historias de usuario y sus tareas correspondiente. En la Tabla 11 se muestra en detalle el plan a seguir en la consecución del producto final.

Sprint	HdU	Tareas	Objs*	Duración Sprint (semanas)
0		T1. Identificar los roles del sistema. T2. Listar historias de usuario. T3. Estimación de historias de usuario. T4. Realización del Anteproyecto	O.1	1
1	HdU1	T1. Decidir el sistema gestor de base de datos óptimo para que la integración con los lenguajes de programación utilizados en el proyecto sea la mejor posible. T2. Modelado de la base de datos y las tablas para almacenar a los usuarios. T3. Realizar conexión entre aplicación y base de datos. T4. Implementación de la funcionalidad registrar usuario. T5. Implementación de la funcionalidad iniciar sesión. T6. Implementación de la funcionalidad error iniciar sesión	O.1 y O.3	4
2	HdU3	T1. Decidir que lenguaje utilizar para el script que genere las lecturas de los contadores. T2. Estudio de los tipos de fraude que se pueden cometer. T3. Creación del script que simule la generación de lecturas de contadores de agua. T4. Implementación del script del primer análisis de fraude. T5. Implementación del script del segundo análisis de fraude. T6. Implementación del script del tercer análisis de fraude. T7. Integrar los scripts con la aplicación.	O.2	4
3	HdU4	T1. Implementación de los cálculos que van a determinar que contadores deben aparecer. T2. Implementación de la funcionalidad obtener contadores.	O.3	3
4	HdU5	T1. Generar varios pdf's que tengan el contenido de los resultados de los análisis y de los contadores con sus respectivas direcciones.	O.4	2
5	HdU2	T1. Implementación para la posterior creación de un proyecto en Big Data Discovery. T2. Integración del script en la aplicación.	O.5	2
6	Fin	T1. Obtener una versión entregable del sistema T2. Realización de la documentación del TFG T3. Realización de un manual de usuario.	-	1

Tabla 11. *Estimación del Plan de Proyecto*

5.1.7. Arquitectura del sistema.

En este sprint inicial se diseña la arquitectura que se pretende desarrollar con la ejecución de este proyecto.

La arquitectura del sistema está basada en una arquitectura cliente-servidor, donde se sigue el patrón Modelo Vista Controlador (MVC). La aplicación desarrollada con el framework y bibliotecas Shiny, Shinydashboard de R, permite el uso de este patrón, pero para ello se ha tenido que escoger la mejor forma para estructurar el proyecto, ya que Shiny permite el desarrollo de dos maneras distintas:

1. Se puede desarrollar toda la funcionalidad en un único archivo llamado ***app.R*** en el que el la interfaz del usuario y el servidor están implementados dentro de ese archivo.
2. Se puede dividir su funcionalidad en dos partes. Por un lado proporcionando la vista de la interfaz de usuario a través de un módulo denominado ***ui.R*** y por otro lado la generación de la funcionalidad se recoge en un módulo llamado ***server.R.***

Por lo tanto, la segunda opción es la utilizada en este TFG, puesto que cumple con el patrón MVC. La Figura 4 muestra cómo sería la arquitectura del sistema

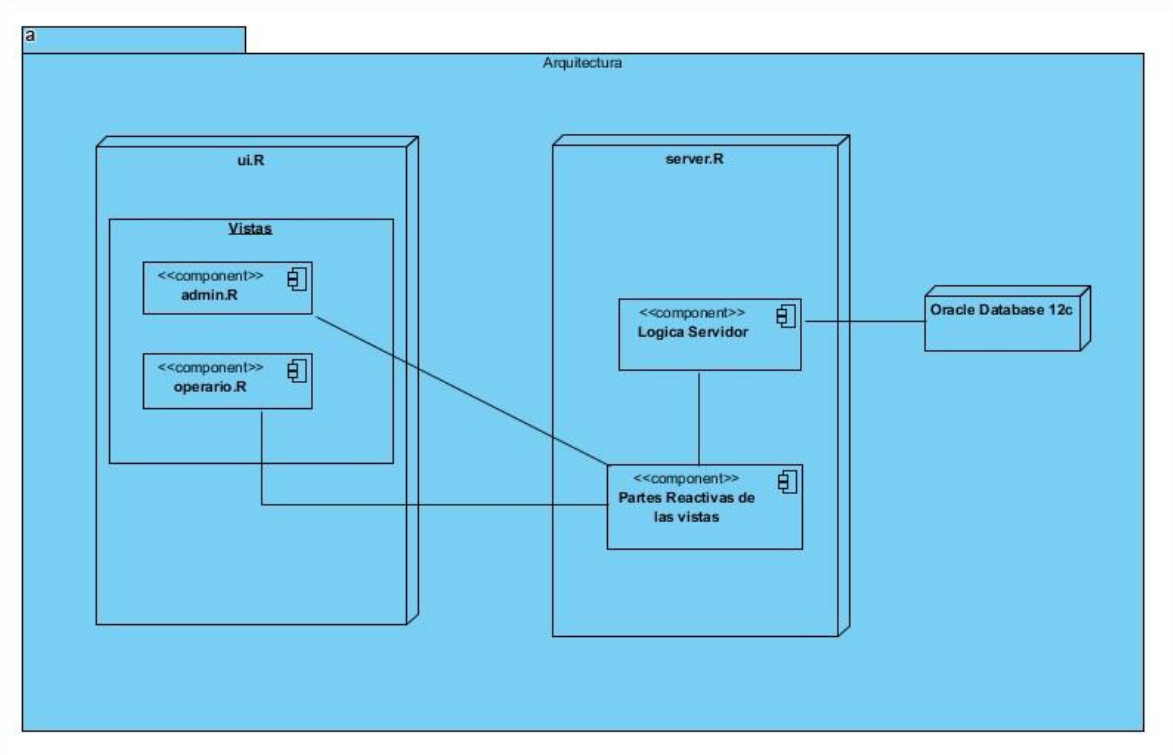

Figura 4. Arquitectura del sistema

5.1.8. Gestión de riesgos.

Para la gestión de los riesgos se ha decidido utilizar una lista de comprobaciones de los riesgos más comunes que surgen en el desarrollo de un proyecto [35]. La Tabla 12. Análisis de riesgos se muestran estos riesgos agrupados por categorías.

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
B. **Organización y gestión**		
B.1. El proyecto languidece en el inicio difuso	10%	3
C. **Ambiente/Infraestructura de desarrollo**		
C.5. Las herramientas de desarrollo no funcionan como esperaban.	30%	5
D. _____		
E. **Cliente**		
E.4. Tiempo de comunicación más lento de lo esperado.	5%	2
F. _____		
G. **Requisitos**		
G.2. Requisitos no definidos correctamente	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

Tabla 12. *Análisis de riesgos*

Habiendo estudiado estos riesgos, se puede concluir que al tratarse de un proyecto en el que solo hay un desarrollador, es muy poco probable que se produzca alguno de ello, no obstante, la estimación del plan de proyecto siempre deja lugar a margen por si surgiese alguna complicación.

5.1.9. Revisión del Sprint.

Una vez finalizado el sprint se evalúa como ha sido la estimación en horas del trabajo que se iba a realizar, frente a las horas que realmente han sido necesarias para su ejecución, así como el aumento de tareas en el caso de ser necesario.

El trabajo real se describe en la Tabla 13.

Sprint	Lista de Tareas	Horas realizadas	Horas estimadas
0	T1. Definición del Proyecto.	15	-
	T2. Identificar los roles del sistema.	2	2
	T3. Listar historias de usuario.	17	10
	T4. Estimación de las historias de usuario.	4	3
	T5. Creación del documento Anteproyecto	35	29,5
	Número total de horas:	**73**	**44,5**

Tabla 13. Revisión del Sprint 0.

Como ya se ha mencionado en el capítulo anterior, los gráficos Burn-up son una herramienta clave para ver el avance, el seguimiento y mostrar de una manera gráfica y visual cual ha sido el trabajo estimado y el trabajo que realmente se ha realizado. En la Figura 5 se muestra esta comparativa:

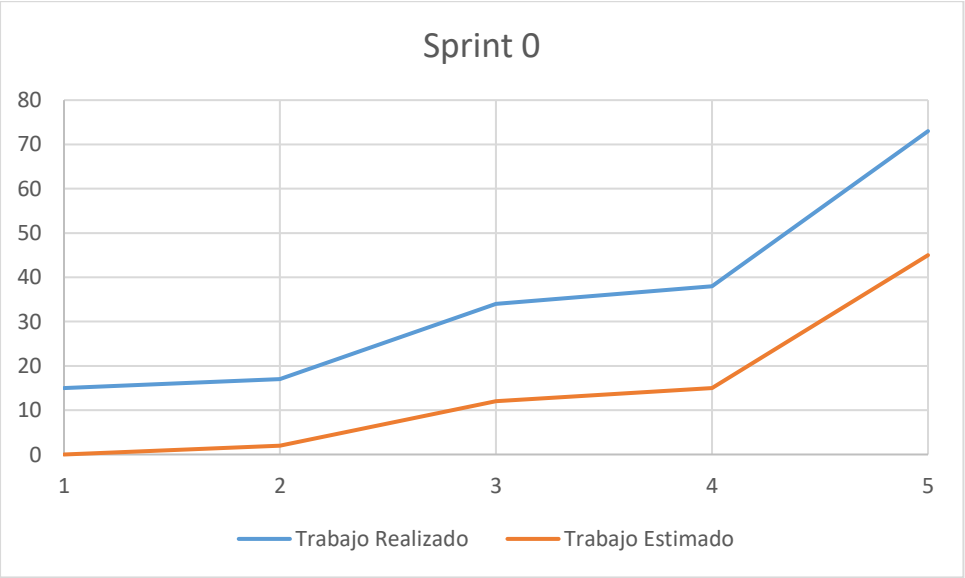

Figura 5. Grafico Burn-Up del Sprint 0.

Como se aprecia en el gráfico se estimó con una vista demasiado optimista la duración de este sprint, puesto que ha sido mucho mayor el trabajo realizado respecto al trabajo estimado. Esto fue debido a la dificultad del alcance del proyecto.

5.1.10. Retrospectiva del Sprint.

Para dar por finalizado el sprint, se reunió el Equipo Scrum para revisar los objetivos marcados en la pila de producto. En este caso se habló de la planificación del proyecto y de la mala planificación del sprint 0, haciendo que se sea más prudente en los demás sprints.

5.2. Sprint 1.

5.2.1. Refinamiento de la pila de producto.

Debido a la pésima planificación realizada en el sprint 0, la pila de producto se ha visto alterada en la duración de las historias de usuario. Este ajuste se ve reflejado en el siguiente apartado.

5.2.2. Reajuste planificación del proyecto.

Como se ha dicho en el apartado anterior, la pésima planificación del sprint 0 hace que se altere toda la planificación de los demás sprint, ya que si no se hace este ajuste sería muy difícil que el producto final esté listo en la fecha acordada. La Tabla 14 muestra cómo quedaría la planificación del proyecto.

Sprint	HdU	Tareas	Objs*	Duración Sprint (semanas)
0	Ini	T1. Definición del Proyecto. T2. Identificar roles del sistema T3. Listar historias de usuario. T4. Estimación de historias de usuario. T5. Realización del Anteproyecto	-	2
1	HdU1	T1. Decidir el sistema gestor de base de datos óptimo para que la integración con los lenguajes de programación utilizados en el proyecto sea la mejor posible. T2. Modelado de la base de datos y las tablas para almacenar a los usuarios. T3. Realizar conexión entre aplicación y base de datos. T4. Implementación de la funcionalidad registrar usuario. T5. Implementación de la funcionalidad iniciar sesión. T6. Implementación de la funcionalidad error iniciar sesión.	O.1 y O.3	5
2	HdU3	T1. Decidir que lenguaje utilizar para el script que genere las lecturas de los contadores. T2. Estudio de los tipos de fraude que se pueden cometer. T3. Creación del script que simule la generación de lecturas de contadores de agua. T4. Implementación del script del primer análisis de fraude. T5. Implementación del script del segundo análisis de fraude. T6. Implementación del script del tercer análisis de fraude. T7. Integrar los scripts con la aplicación.	O.2	3
3	HdU4	T1. Implementación de los cálculos que van a determinar que contadores deben aparecer T2. Implementación de la funcionalidad obtener contadores	O.3	2
4	HdU5	T1. Generar varios pdf's que tengan el contenido de los resultados de los análisis y de los contadores con sus respectivas direcciones.	O.4	2
5	HdU2	T1. Implementación para la posterior creación de un proyecto en Big Data Discovery. T2. Integración del script en la aplicación.	O.5	1
6	Fin	T1. Obtener una versión entregable del sistema T2. Realización de la documentación del TFG T3. Realización de un manual de usuario.	-	1

Tabla 14. *Reajuste del Plan de Proyecto.*

Finalizada ya la reestimación, se pasa a desarrollar la historia de usuario que corresponde a este sprint.

5.2.3. Planificación del Sprint.

En este caso la pila de sprint está formada únicamente por la historia de usuario 1. Véase Tabla 15.

Historia de Usuario	
Número: 1	**Usuario:** Analista de fraude y Operario
Nombre de la historia: Gestión de los roles de usuario	
Prioridad de negocio: 85	
Esfuerzo: 144	**Sprint asignado:** 1
Programador responsable: Francisco Parreño Heredia	
Descripción: Como usuario quiero poder registrarme e identificarme en la aplicación según mi rol que desempeñe en la organización.	
Precondición: No es necesaria ninguna precondición para la realización de la historia de usuario.	
Postcondición: Tras la realización de este esta historia de usuario, el usuario podrá identificarse con su correspondiente rol.	
Tareas (T): T1. Decidir el sistema gestor de base de datos óptimo para que la integración con los lenguajes de programación utilizados en el proyecto sea la mejor posible. T2. Modelado de la base de datos y las tablas para almacenar a los usuarios. T3. Implementación de la funcionalidad registrar usuario. T4. Implementación de la funcionalidad loguear usuario. T5. Implementación de la funcionalidad error al loguear.	
Artefacto o producto generado: • Funcionalidad login usuarios. • Funcionalidad login de usuarios erróneo. • Funcionalidad registrar usuarios.	

Tabla 15. Historia de Usuario del Sprint 1.

Se va a desarrollar la historia de usuario de gestión de los roles de usuario. En la Tabla 16 e puede observar las tareas derivadas de esta historia de usuario y la estimación del trabajo que se debe realizar para cada una de ellas. Se puede apreciar también como alguna de las

tareas se han dividido en tareas más pequeñas o subtareas las cuales han sido estimadas de la misma manera que las tareas.

Sprint	Tareas	Objs*	Horas estimadas
1	T1. Configurar entorno de desarrollo	O.1 y O.3	-
	T1.1. Instalación de la máquina virtual Big Data Lite 4.3.1		1
	T1.2. Instalación de R.		0,5
	T1.3. Instalación de RStudio.		0,75
	T1.4. Instalación del paquete Shiny.		0,75
	T1.5. Instalación del paquete Shinydashboard.		0,5
	T1.6. Instalación del paquete ROracle.		0,75
	T1.7. Instalación del paquete ORCH.		0,75
	T1.8. Creación de un repositorio en Bitbucket		1
	T1.9. Conectar repositorio con RStudio.		0,25
	T2. Estudio de las nuevas tecnologías.		-
	T2.1. Aprendizaje de R.		39
	T2.2. Aprendizaje de Shiny y Shinydashboard.		39
	T3. Decidir el sistema gestor de base de datos relacional óptimo para que la integración con los lenguajes de programación utilizados en el proyecto sea la mejor posible.		5
	T3.1. Instalación de la base de datos Oracle 12c.		3
	T3.2. Instalación de Oracle SQLDeveloper.		0,5
	T4. Modelado de la base de datos y las tablas para almacenar a los usuarios.		2
	T4.1. Conexión con la base de datos.		0,25
	T5. Implementación de la funcionalidad registrar usuario.		22
	T6. Implementación de la funcionalidad iniciar sesión		15
	T7. Implementación de la funcionalidad error al iniciar sesión		12
		Número total de horas:	**144**

Tabla 16. *Tareas estimadas del Sprint 1.*

5.2.4. Desarrollo de tareas.

5.2.4.1. T.1. Configurar entorno de desarrollo.

Para la realización de este TFG en primer lugar se ha llevado a cabo la configuración del entorno de desarrollo, ya que la tecnología Big Data hace que este proceso sea vital para los posteriores desarrollos. El TFG se realiza en un equipo con las siguientes características:

- Procesador: Intel® Core™ i5-3340 CPU @ 2.70GHz
- Memoria RAM: 16 GB
- Tipo de sistema: Sistema Operativo 64 bits
- Sistema operativo nativo: Windows 7, aunque la finalización del proyecto se ha llevado acabo con Windows 10.
- El entorno Big Data se ha llevado acabo sobre una máquina virtual con Oracle Linux 6.7 llamada Big Data Lite 4.3.1, en la que está instalado Hadoop como un único nodo.

En primer lugar, se descarga la máquina virtual desde la página principal de Oracle y se importa en el software de virtualización Oracle VM Virtual Box. A partir de este punto se desarrollarán las siguientes subtareas:

- Instalación de R y R Studio: se realiza mediante el siguiente comando introducido por el terminal. Listado 1.

```
wget https://download2.rstudio.org/rstudio-server-rhel-0.99.489-x86_64.rpm

sudo yum install -y --nogpgcheck rstudio-server-rhel-0.99.489-x86_64.rpm
```

Listado 1. Instalación R y R Studio.

- Instalación de los paquetes Shiny, Shinydashboard para la creación de la aplicación web: dentro de la consola de RStudio se ejecuta las líneas de código que se muestran en el Listado 2.

```
install.packages ("shiny")

install.packages ("shinydashboard")
```

Listado 2. Instalación de paquetes Shiny y Shinydashboard.

- Instalación de los paquetes R Oracle y ORCH: estos paquetes se instalan de la misma forma que los dos anteriores. Son unos paquetes para establecer la conexión entre R y la base de datos y entre R y Hadoop, que es en su sistema de ficheros distribuidos (HDFS) donde se van a guardar los resultados de los análisis de detección de fraude. Listado 3

```
install.packages ("ROracle")

install.packages ("ORCH")
```

Listado 3. *Instalación de paquetes ROracle y ORCH.*

- Creación un repositorio Bitbucket para el seguimiento del estado del desarrollo del proyecto y del control de versiones con Git.

5.2.4.2. T.3. Decidir Sistema gestor de base de datos relacional.

Una vez se tiene la configuración del entorno de desarrollo preparada, toca decidir qué sistema gestor de base de datos utilizar. Primero señalar que los datos que se van a almacenar en la base de datos son:

- Se guardarán el nombre, contraseña y rol de los usuarios que accedan al sistema por medio de la aplicación web.
- Se deberá almacenar también las direcciones en las que se encuentran los contadores de agua para que así el operario pueda personarse y comprobar la anomalía o anomalías de las lecturas.

Al estar programando con R se debe buscar una solución que se integre perfectamente con dicho lenguaje. R tiene un amplio abanico de paquetes que nos ofrece poder conectar con bases de datos sin que esto sea un auténtico quebradero de cabeza.

Cabe destacar que como se ha mencionado al inicio de este documento, el TFG se está realizando con tecnología Oracle debido a que la empresa *avanttic consultoría tecnología* es Partner Platinium de Oracle, por lo que se ha intentado siempre que ha sido posible la utilización de dicha tecnología en este proyecto.

Con lo anteriormente expuesto se ha decidido utilizar la base de datos de Oracle 12c debido a que la instalación del paquete en R "ROracle" citado en el apartado anterior nos ofrece esa conectividad con la base de datos. La conexión puede verse en el Listado 4.

```
require(ROracle)

drv <- dbDriver("Oracle")
connection <- dbConnect(drv,
                        user=”SH_APP”,
                        password=”SH_APP”,
                        dbname=”localhost:1521/orcl”
)
```

Listado 4. Conexión con la base de datos de los usuarios.

5.2.4.3. T.4. Modelado de la base de datos y las tablas para almacenar a los usuarios.

Para el inicio de la aplicación web hace falta tener un rol específico para poder acceder a ella, ya que así se evita la entrada de cualquiera en el sistema. El diseño que se ha utilizado para controlar la seguridad son 3 tablas:

- La primera tabla corresponde a los usuarios cuyos atributos son nombre y contraseña.
- La segunda tabla corresponde a los roles que puede tomar ese usuario cuyos atributos son id rol y nombre del rol.
- La tercera tabla es producto de la asociación de las dos anteriores con esta, cuyos atributos son: el nombre del usuario y el id del rol.

Este diseño tiene la ventaja de poder añadir algún rol más de forma sencilla, y no permite la eliminación del usuario por error a menos que desaparezca de la tercera tabla mencionada anteriormente. En la Figura 6 se muestra el modelo de datos resultante:

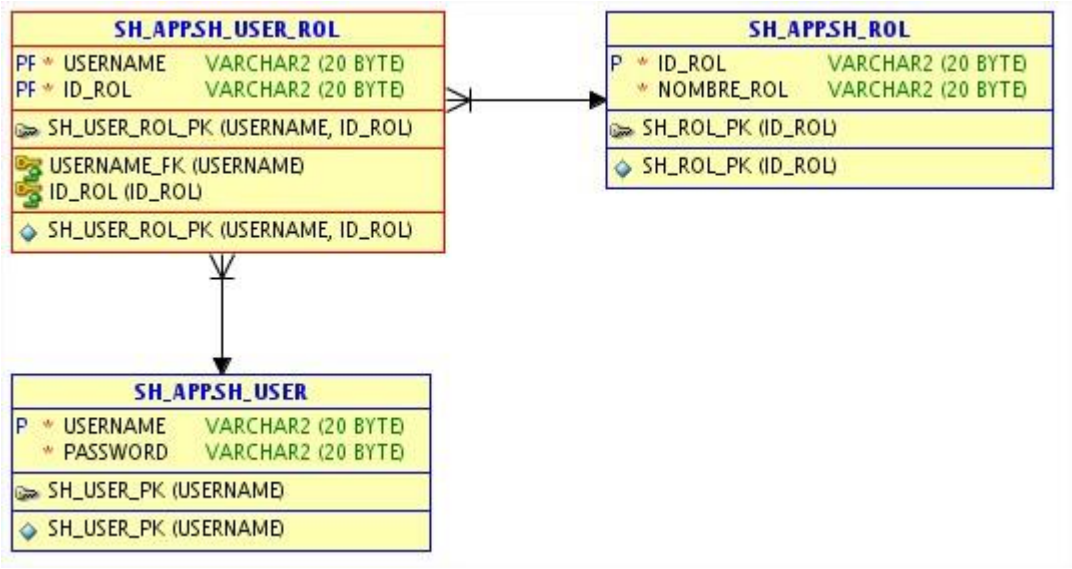

Figura 6. Modelo de la base de datos de los Usuarios

5.2.4.4. T.5. Implementación de la funcionalidad registrar usuario.

Antes de pasar a la implementación se va a mostrar un boceto creado con Balsamiq Mockups para ver cuál sería el resultado de esta implementación. Se muestra en la Figura 7.

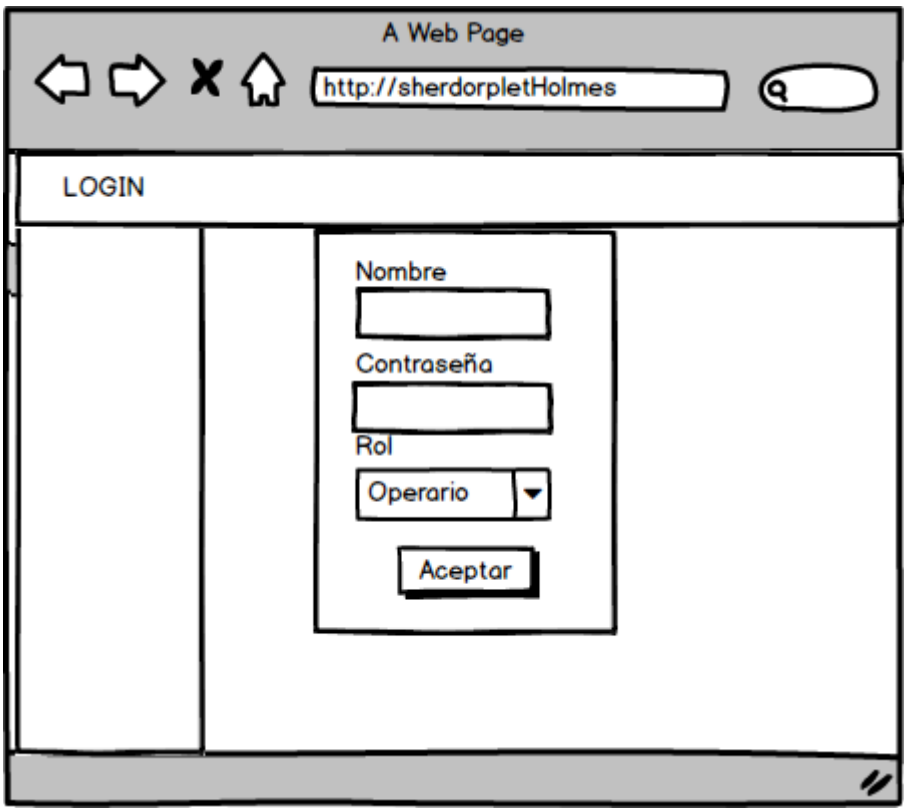

Figura 7. Boceto de la Tarea T5 del Sprint 1.

Como se ha dicho en el apartado 5.1.7 de la arquitectura del sistema, la implementación se va a llevar a cabo con el entorno de desarrollo R Studio y con el paquete Shiny y Shinydashboard para la creación de una interfaz con la estructura de un dashboard. Dicho esto, se van a crear tantos ficheros como vistas tenga la aplicación. Para la implementación de registrar usuarios el código a desarrollar se encuentran los ficheros ***server.R***, ***ui.R*** y ***resgister.R***.

El fichero ***ui.R*** es el modulo por defecto que va a renderizar cada una de las vistas que se le indique por el servidero, en este caso la vista que renderizará será la de ***register.R.***

El Listado 5 muestra el código creado para ***ui.R***

```
library (shiny)
library (shinydashboard)

titleRole<-textOutput("title")

shinyUI(
  dashboardPage(
    dashboardHeader(title=titleRole, menuItemOutput("header")),
    dashboardSidebar(uiOutput("side")),
    dashboardBody(uiOutput("page"))
  )
)
```

Listado 5. *Código del archivo ui.R*

Una vez implementado el código de ***ui.R*** se implementará la vista ***register.R*** que estará compuesta por el título, el menú lateral izquierdo y la parte de central de la página en la que se encuentra el módulo de registrar.

El Listado 6 muestra el código creado para ***register.R***

```
register_title="REGISTER"

register_side=(sidebarMenu())

register_main=
  fluidRow(
    column(7,wellPanel(textInput("userName", "Username"),
                  passwordInput("passwd", "Password"),
                  selectInput("select", "Selecciona Rol",
                              choices = list("Analista de fraude" = 1,
"Operario" = 2),
                              selected = 1),
                  br(),hr(),actionButton("Aceptar", "Aceptar"))))
```

Listado 6. *Código archivo register.R*

Para terminar la vista y la funcionalidad de registrar el usuario en la base de datos, se tiene que implementar todo ese código en la parte del servidor llamado ***server.R.***

Se crea un *observeEvent* el cual responde a eventos y únicamente se activa cuando el usuario ha introducido el nombre, contraseña y su rol. El Listado 7 muestra el código de este "observeEvent"

```
observeEvent(input$Aceptar,{
      switch (input$Aceptar,
               "1"={
                 nombreRegistrado<-renderText({input$userName})
                 pass<-renderText({input$passwd})
                 queryUserPass<-paste("insert into SH_USER (USERNAME,PASSWORD)
values('",nombreRegistrado(),"','","'",pass(),"')", sep="")
                 dbSendQuery(connection, queryUserPass)
                 queryRol<-paste("insert into SH_USER_ROL (USERNAME,ID_ROL)
values('",nombreRegistrado(),"','","'1')", sep="")
                 dbSendQuery(connection, queryRol)
                 dbCommit(connection)
               },
              "2"={
                nombreRegistrado<-renderText({input$userName})
                pass<-renderText({input$passwd})
                queryUserPass<-paste("insert into SH_USER (USERNAME,PASSWORD)
values('",nombreRegistrado(),"','","'",pass(),"')", sep="")
                dbSendQuery(connection, queryUserPass)
                queryRol<-paste("insert into SH_USER_ROL (USERNAME,ID_ROL)
values('",nombreRegistrado(),"','","'2')", sep="")
                dbSendQuery(connection, queryRol)
                dbCommit(connection)
              }
      )
    })
```

Listado 7. *Código crear usuario.*

Se puede observar que dentro de la función observeEvent se establece la conexión con la base de datos y se introducen en las tablas de usuarios y usuario-rol.

Finalmente, la Figura 8 muestra la vista del módulo registrar, con toda su funcionalidad en la parte del servidor, una vez terminada la implementación de esta tarea.

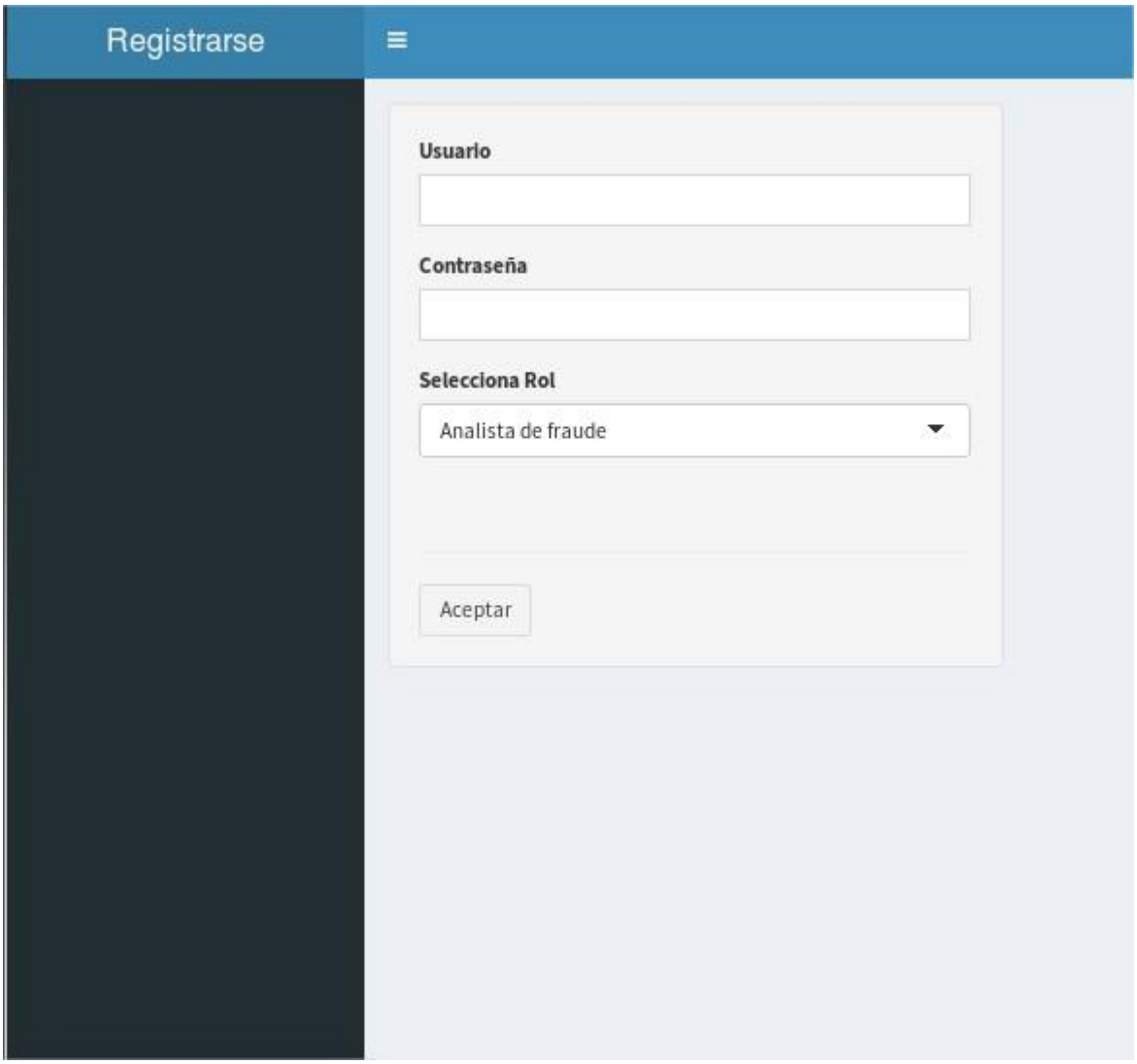

Figura 8. Vista de la aplicación: Registrarse.

5.2.4.5. T.6. Implementación de la funcionalidad iniciar sesión.

En esta tarea se implementará la funcionalidad de acceder al sistema mediante el nombre de usuario y la contraseña.

Como en la anterior tarea, se va a mostrar en la Figura 9 un boceto de cómo quedaría la implementación de esta funcionalidad.

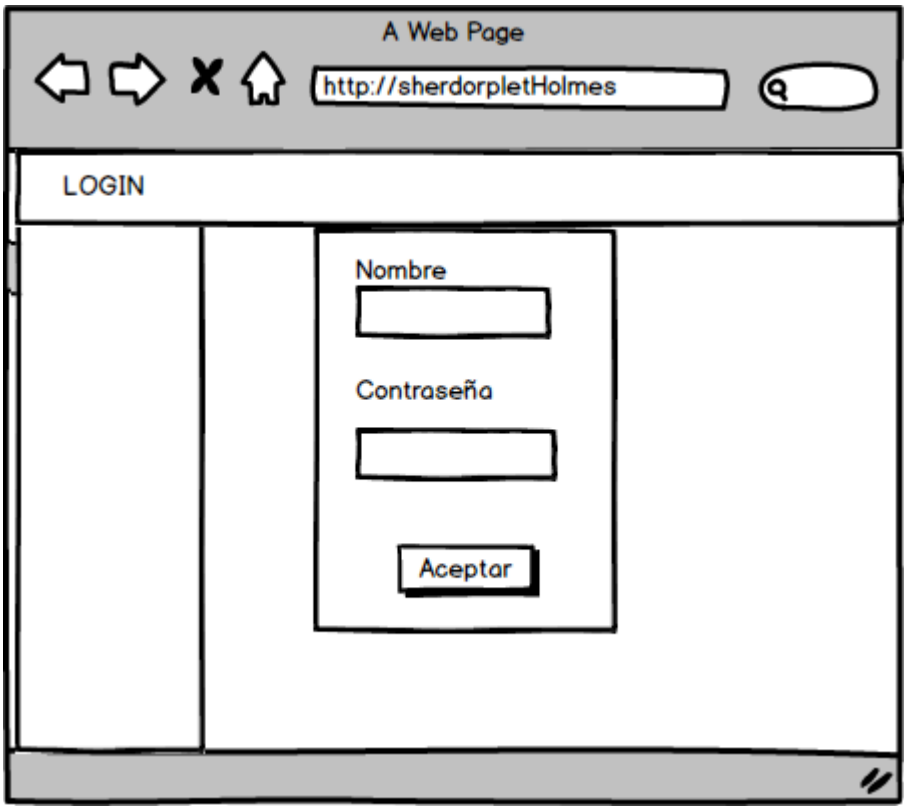

Figura 9. Boceto de la Tarea T6 del Sprint 1.

La aplicación tiene que identificar que usuario está accediendo a ella para ello se realizan las siguientes operaciones internas:

1. Se comprueba que exista ese usuario con esa contraseña.
2. Si existe ese usuario, se busca en la tabla usuario-rol el rol que desempeña en la organización.
3. Dependiendo del rol se genera la vista del analista de fraude o la vista del operador.

El Listado 8 muestra el código para realizar estas comprobaciones:

```
shinyServer(
  function(input, output,session) {

    USER <- reactiveValues(Logged = FALSE,role="0")
    observe({
      if (USER$Logged == FALSE) {
        if (!is.null(input$Login)) {
          if (input$Login > 0) {
            username_input <- isolate(input$userName)
            password_input <- isolate(input$passwd)

            userpass<-fetch(dbSendQuery(connection, "select USERNAME,PASSWORD
from SH_USER"))
            user_role<-fetch(dbSendQuery(connection, "select USERNAME, ID_ROL
from SH_USER_ROL"))

            for (i in 1:length(userpass$USERNAME)) {
              if ((userpass$USERNAME[i] == username_input) &
(userpass$PASSWORD[i] == password_input)){
                USER$Logged <<- TRUE
                for (j in 1:length(user_role$USERNAME)) {
                  if (user_role$USERNAME[j] == userpass$USERNAME[i]){
                    USER$role <<- user_role$ID_ROL[j]
                  }
                }
              }
            }
          }
        }
      }
    })
```

Listado 8. *Código para la comprobación del rol del usuario.*

Se utiliza una variable reactiva llamada USER para almacenar el nombre, contraseña y rol de ese usuario.

La programación reactiva se caracteriza por enfatizar el uso de valores que cambian con el tiempo y de expresiones que registran esos cambios, esto quiere decir que la variable USER cuando ya ha accedido a la aplicación cambia sus valores iniciales por unos nuevos valores de *USER$Logged* igual a TRUE y *USER$role* igual al rol que tenga dicho usuario.

La Figura 10 muestra la vista del módulo login, con toda su funcionalidad en la parte del servidor, una vez terminada la implementación de esta tarea.

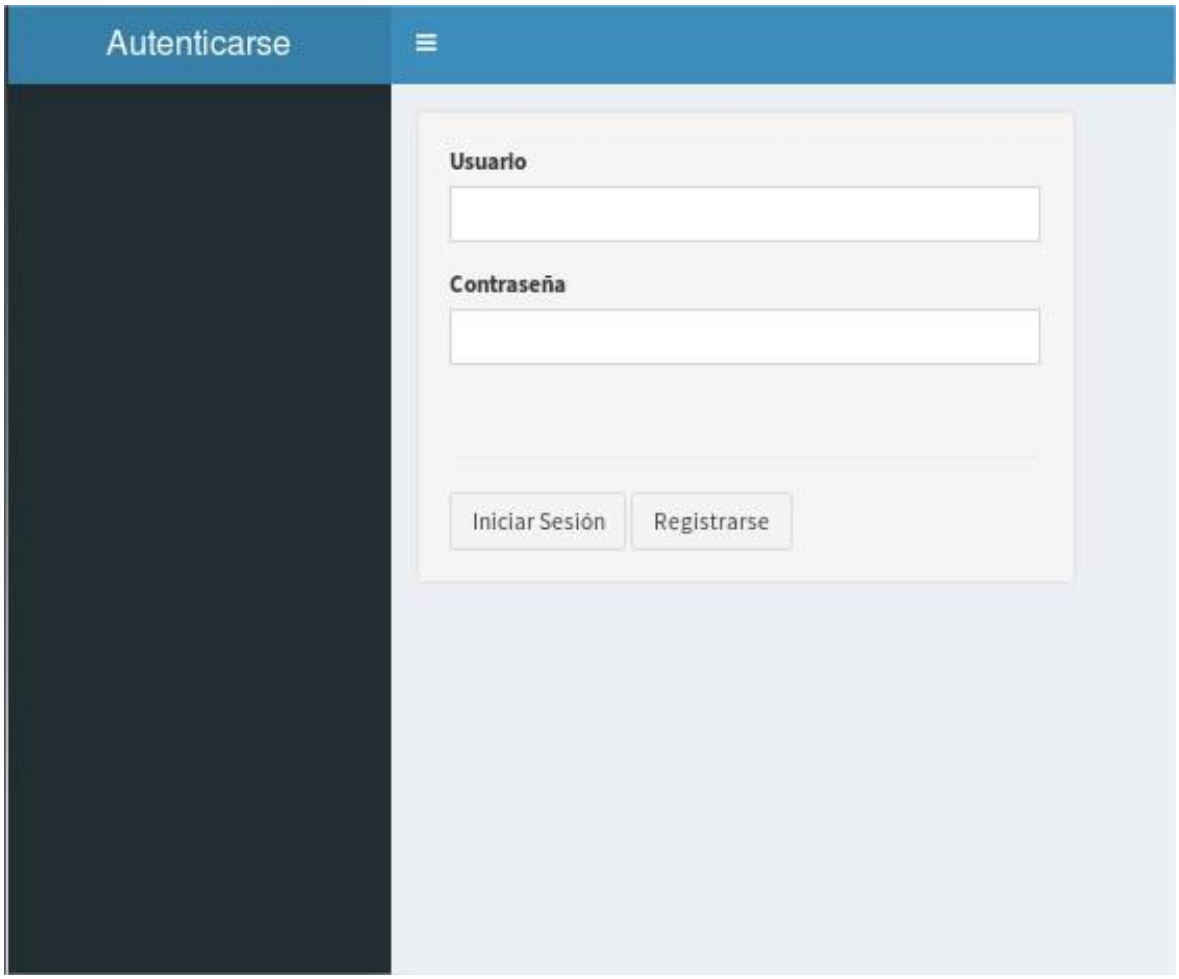

Figura 10. *Vista de la aplicación: Autenticarse.*

5.2.4.6. T.7. Implementación de la funcionalidad “error al hacer login”.

La implementación de esta última tarea del sprint está relacionada con la tarea anterior. Se comprueba que el usuario esté dentro de la base de datos del sistema, en este caso la comprobación no obtendrá el nombre ni la contraseña ya que estos datos no se encuentran en la base de datos por lo que aparecerá un mensaje de error al usuario indicando que el usuario o la contraseña son incorrectos. Esto llevará de nuevo al usuario a la ventana de *Autenticarse*.

5.2.5. Revisión del Sprint.

Al finalizar este Sprint se produjo su revisión. Por parte del desarrollador se destacó el fuerte aprendizaje que llevo a cabo en el estudio de R y Shiny, este aprendizaje ocupó gran parte

de la realización del sprint, pero era necesario para que en futuros sprints el tiempo de implementación fuese menor.

La Tabla 17 muestra la comparativa de horas estimadas y horas realizadas tras la finalización de este Sprint.

Sprint	Tareas	Horas realizadas	Horas estimadas
1	T1. Configurar entorno de desarrollo	-	-
	T1.1. Instalación de la máquina virtual Big Data Lite 4.3.1	1	1
	T1.2. Instalación de R.	0,25	0,5
	T1.3. Instalación de R Studio.	0,25	0,5
	T1.4. Instalación del paquete Shiny.	0,25	0,5
	T1.5. Instalación del paquete Shinydashboard.	0,25	0,5
	T1.6. Instalación del paquete ROracle.	0,25	0,5
	T1.7. Instalación del paquete ORCH.	0,25	0,5
	T1.8. Creación de un repositorio en Bitbucket	1	1
	T1.9. Conectar repositorio con RStudio.	0,25	0,25
	T2. Estudio de las nuevas tecnologías.	-	-
	T2.1. Aprendizaje de R.	35	40
	T2.2. Aprendizaje de Shiny y Shinydashboard.	32,25	39
	T3. Decidir el sistema gestor de base de datos óptimo para que la integración con los lenguajes de programación utilizados en el proyecto sea la mejor posible.	3	5
	T3.1. Instalación de la base de datos Oracle 12c.	2	2
	T3.2. Instalación de Oracle SQLDeveloper.	0,5	0,5
	T4. Modelado de la base de datos y las tablas para almacenar a los usuarios.	2	2
	T4.1. Conexión con la base de datos.	0,25	0,25
	T5. Implementación de la funcionalidad registrar usuario.	12.5	25
	T6. Implementación de la funcionalidad iniciar sesión	5	15
	T7. Implementación de la funcionalidad error al iniciar sesión	3	10
	Número de total de horas:	**100,5**	**144**

Tabla 17. *Trabajo del Sprint 1.*

En la Figura 11 se muestra de una manera más grafica la comparativa entre las horas estimadas y las horas realizadas. Para mayor claridad se van a mostrar solo las tareas, ya que las subtareas están bien definidas en la tabla anterior.

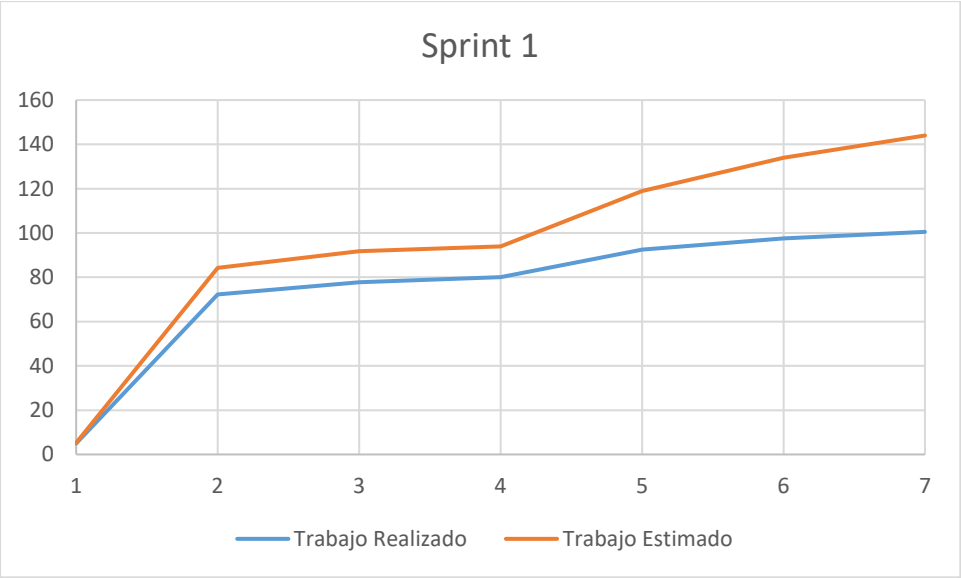

Figura 11. *Gráfico Burn-Up del Sprint 1.*

El sprint 1 ha estado mejor estimado que el anterior, ya que se presuponía que el estudio del lenguaje de programación (desconocido por el desarrollador) iba a abarcar la mayor parte de este sprint.

5.2.6. Retrospectiva del Sprint.

Para finalizar este sprint, se reunió el Equipo Scrum para revisar los objetivos marcados con la pila de producto. Este dio como resultado el primer prototipo del producto, con las funcionalidades de autenticarse y registrarse en la aplicación.

5.3. Sprint 2.

5.3.1. Refinamiento de la pila de producto.

La pila de producto no ha sufrido modificaciones tras su revisión.

5.3.2. Planificación del Sprint.

En este caso la pila de sprint 2 está formada únicamente por la historia de usuario 3. Véase Tabla 18.

Historia de Usuario	
Número: 3	**Usuario:** Analista de fraude
Nombre de la historia: Identificar fugas y fraudes.	
Prioridad de negocio: 95	
Esfuerzo: 89	**Sprint asignado:** 2
Programador responsable: Francisco Parreño Heredia	
Descripción: Como analista de fraude quiero poder identificar los posibles contadores sospechosos de fraude y posibles fugas.	
Precondición: Haber iniciado sesión en la aplicación teniendo el rol de analista de fraude.	
Postcondición: El analista de fraude tendrá en la aplicación los resultados de cada análisis de detección de fraude y fugas.	
Tareas (T): T1. Decidir que lenguaje utilizar para el script que genere las lecturas de los contadores. T2. Estudio de los tipos de fraude que se pueden cometer. T3. Creación del script que simule la generación de lecturas de contadores de agua. T4. Implementación del script del primer análisis de fraude. T5. Implementación del script del segundo análisis de fraude. T6. Implementación del script del tercer análisis de fraude. T7. Integrar los scripts con la aplicación.	
Artefacto o producto generado: • Gráfica para la representación del primer análisis. • Tabla para la representación del segundo análisis. • Tabla para la representación del tercer análisis.	

Tabla 18. Historia de Usuario del Sprint 2.

Se va a desarrollar la historia de usuario de identificar fugas y fraudes en contadores de agua. En la Tabla 19 se puede observar las tareas derivadas de esta historia de usuario y la estimación del trabajo que se debe realizar para cada una de ellas. Se puede apreciar también como alguna de las tareas se han dividido en tareas más pequeñas o subtareas las cuales han sido estimadas de la misma manera que las tareas.

Dado que la estimación del sprint 1 con respecto a las tareas del aprendizaje del lenguaje se hizo de una manera pesimista, el desarrollador adquirió un conocimiento del lenguaje que le permite realizar las funcionalidades de manera más rápida y eficaz debido al tiempo que empleó en el estudio. Esto hace que la estimación de los siguientes sprints sea más fiable.

Sprint	Tareas	Objs*	Horas estimadas
2	T1. Decidir que lenguaje utilizar para el script que genere las lecturas de los contadores.	O.2	3
	T2. Estudio de los tipos de fraude que se pueden cometer.		18
	T3. Creación del script que simule la generación de lecturas de contadores de agua.		-
	T3.1. Obtener una base de datos de inmuebles fiable.		4
	T3.2. Clasificar los inmuebles según sus características.		6
	T3.3. Generar los consumos de agua habiendo introducido fraude en ellos.		12
	T4. Implementación del script del primer análisis de fraude.		22
	T5. Implementación del script del segundo análisis de fraude.		10
	T6. Implementación del script del tercer análisis de fraude.		10
	T7. Integrar los scripts con la aplicación.		4
		Número total de horas:	**89**

Tabla 19. *Tareas estimadas del Sprint 2.*

5.3.3. Desarrollo de Tareas.

5.3.3.1. T1. Decidir lenguaje script generador de lecturas.

La elección del lenguaje del script que genere las lecturas de los contadores es muy importante. Si se quisiera ampliar el alcance de este proyecto, se debería utilizar un lenguaje que tenga compatibilidades con R (librerías, paquetes, etc…) puesto que R es el lenguaje principal de este TFG. Además de tener una curva de aprendizaje sencilla para no ampliar el tiempo de entrega del producto.

Expuesto lo anterior, se ha elegido Python como lenguaje para escribir el script que permite simular los consumos de los contadores de agua.

5.3.3.2. T2. Estudio de los tipos de fraude y fugas en contadores de agua.

Antes de pasar a explicar la creación del script para generar los consumos de los contadores, se debe saber que fraudes se cometen en la actualidad para que a la hora de simular los consumos se pueda introducir fraude en ellos y así los algoritmos de detección sean capaces de identificar que contadores son sospechosos de fraude.

Como se ha explicado en el Capítulo 3 se ha realizado un estudio anterior para averiguar esta problemática. Se ha llegado a la conclusión de que los contadores sospechosos de fraude pueden ser detectados por estas consecuencias:

1. Por una caída progresiva en el consumo.
2. Una caída súbita en el consumo y normalización a partir de ahí.
3. Un consumo anual anormalmente bajo.

La implementación de estos tres análisis se realizará en este sprint ya que corresponden a las tareas 4, 5 y 6 de dicho sprint.

5.3.3.3. T3. Creación del Script para generar los consumos de los contadores.

Esta tarea está dividida en tareas más pequeñas, ya que se necesita preparar los datos y otras opciones para generar los consumos.

- *T3.1. Obtener una base de datos de inmuebles fiable.*

  Cada contador debe estar asociado con una dirección, para ello se ha cogido una base de datos de la sede del catastro de la comunidad de Madrid. Esta base de datos es accesible a cualquier ciudadano aportando el documento nacional de identidad.

- *T3.2. Clasificar los inmuebles según sus características.*

  Debido a que los datos obtenidos de la subtarea anterior están en bruto, es decir, con muchos atributos que no sirven a la hora de clasificarlos, llega el momento de seleccionar aquellos atributos que ayuden a clasificarlo. Para este caso los atributos seleccionados son: superficie construida y número de habitantes por inmueble.

El Listado 9 muestra la llamada principal para realizar la clasificación según estas dos variables.

```
require(cluster)

cluster<-clara (datos_2_variables, 5)
```

Listado 9. Código parcial para la realización de los clusters.

Con las dos subtareas anteriores completadas y sabiendo las técnicas para detectar los fraudes, se pasó a la creación del generador de consumos. Para la simulación de unos consumos fiables, se han consultado varios datos realizados por el Instituto Nacional de Estadística *(INE)*. Estos datos son:

- El consumo de agua por persona.
- Porcentaje de personas que habitan en una vivienda por metro cuadrado.
- Consumo medio de cada vivienda en las distintas estaciones del año.
- Consumo de otros inmuebles a parte de las ya mencionadas viviendas.

Dados todos los pasos previos se obtiene la generación de unos datos en los que se ha introducido fraudes para que luego sean detectados y el sistema cumpla con el objetivo. En el Listado 10 se muestra un ejemplo del formato que tienen los datos una vez simulados.

```
ID_CONTADOR,IND_LEC,FECHA_LEC,CLUSTER,CONSUMO
58094603,331.919,2014-01-01 20:01:14,5,4.585
```

Listado 10. Salida de las lecturas de los contadores.

5.3.3.4. T4. Implementación del script del primer análisis de fraude.

La implementación de esta tarea se ha basado en el coeficiente de correlación de Pearson que como bien se explica en el Capítulo 3, es una medida de la relación lineal entre dos variables aleatorias cuantitativas. Las dos variables medidas son el tiempo y en consumo.

El Listado 11 muestra las funciones MapReduce para la obtención de estos resultados.

```
res_1Fraude <- hadoop.run(
  sfo.dfs.part,
  mapper = function(key, ontime) {
    for (i in seq_len(length(key))){
      orch.keyval(key[i], ontime[i,])
    }
  },
  reducer = function(key, vals) {
    consumo <- c()
    tiempo<- c()
    for (x in 1:nrow(vals)) {
      tiempo<-c(tiempo,x)
      consumo <- c(consumo,vals$V5[x])
    }
    orch.keyval(key, cor(tiempo, consumo))
  }
)
```

Listado 11. *Funciones MapReduce para el primer análisis.*

La función reducer obtiene una serie de valores (clave, valor) de la función mapper y muestra como resultado el código del contador y el coeficiente de correlación de Pearson, cuyos valores están comprendidos entre -1 y 1. El Listado 12 muestra un ejemplo de la salida de la función reducer.

```
38005133        -0.939336436627724
38142427        0.424264068711929
38377634        0.802288821021566
38505136        -0.894427190999916
```

Listado 12. *Salida de la función reducer del primer análisis.*

5.3.3.5. T5. Implementación del script del segundo análisis de fraude.

La implementación de esta tarea se ha basado en dividir las lecturas de un año en cuatro trimestres desde enero hasta diciembre, y se calcula la desviación típica entre dos trimestres adyacentes además del porcentaje de decremento entre estos dos trimestres.

El Listado 13 muestra las funciones MapReduce para la obtención de estos resultados.

```
res_2Fraude <- hadoop.run(
  sfo.dfs.part,
  mapper = function(key, ontime) {
    for (i in seq_len(length(key))){
      orch.keyval(key[i], ontime[i,])
    }
  },

  reducer = function(key, vals) {
    mes<-c()
    primerT<-0; segundoT<-0; tercerT<-0; cuartoT<-0
    dt1y2<-c(); dt2y3<-c(); dt3y4<-c()
    for (x in 1:nrow(vals)) {
      mes<-strsplit(vals$V3[x], "-")[[1]][2]
      if ((mes>='01') && (mes <= '06')){
        dt1y2 <- c(dt1y2,vals$V5[x])
        if ((mes>='01') && (mes <= '03')){
          primerT <- primerT + vals$V5[x]
        }
        if ((mes>='04') && (mes <= '06')){
          segundoT <- segundoT + vals$V5[x]
        }
      }

      if ((mes>='04') && (mes <= '09')){
        dt2y3 <- c(dt2y3,vals$V5[x])
        if ((mes>='07') && (mes <= '09')){
          tercerT <- tercerT + vals$V5[x]
        }
      }

      if ((mes>='07') && (mes <= '12')){
        dt3y4 <- c(dt3y4,vals$V5[x])
        if ((mes>='10') && (mes <= '12')){
          cuartoT <- cuartoT + vals$V5[x]
        }
      }
    }
    porcentaje1y2<-100-((segundoT*100)/primerT)
    porcentaje2y3<-100-((tercerT*100)/segundoT)
    porcentaje3y4<-100-((cuartoT*100)/tercerT)

    resultado<-
data.frame(sd(dt1y2),porcentaje1y2,sd(dt2y3),porcentaje2y3,sd(dt3y4),porcentaje
3y4)
    orch.keyval(key, resultado)
  })
```

Listado 13*. Funciones MapReduce del segundo análisis.*

La función reducer obtiene una serie de valores (clave, valor) de la función mapper y muestra como resultado el código del contador, la desviación del par de trimestres adyacentes y el porcentaje de decremento (en caso de que lo hubiese) de consumo con respecto a dos trimestres. Los resultados están delimitados por ",". El Listado 14 muestra un ejemplo de la salida de la función reducer.

```
38505136     0.013,20.28, 0.023,6.12, 0.407, 11.28
38905113     0.056,25.13, 0.002,3.33, 0.026, 10.23
```

Listado 14*. Salida de la función reducer del segundo análisis.*

5.3.3.6. Implementación del script del tercer análisis de fraude.

La implementación de esta tarea se basa en calcular el consumo anual de cada contador, y si este consumo está por debajo de unos límites, indica que ese contador puede ser sospechoso se fraude.

El Listado 15 muestra las funciones MapReduce para la obtención de estos resultados.

```
res_3Fraude <- hadoop.run(
  sfo.dfs.part,
  mapper = function(key, ontime) {
    cat(class(key))
    cat(class(ontime))
    for (i in seq_len(length(key))){
      orch.keyval(key[i], ontime[i,])
    }
  },
  reducer = function(key, vals) {
    consumoAnual <- 0
    for (x in 1:nrow(vals)) {
      consumoAnual <- consumoAnual + vals$V5[x]
    }
    orch.keyval(key, consumoAnual)
  }
)
```

Listado 15*. Funciones MapReduce del tercer análisis.*

La función reducer obtiene una serie de valores (clave, valor) de la función mapper y muestra como resultado el código del contador y el consumo anual de cada contador.

5.3.3.7. Integrar los scripts en la aplicación web.

Una vez implementados todos los scripts, llega el momento de integrarlos en la aplicación. Cada uno de los scripts está guardado en archivos distintos, por lo que se tiene tres scripts que deben ser cargados dentro del archivo ***server.R***. Como no es necesario que estén cargados todos los scripts desde el principio se va a crear un evento para que cargue el análisis correspondiente. El Listado 16 muestra la forma en la que se carga uno de los scripts.

```
observeEvent(input$PrimerFraude,{
      dirr<-renderText({ input$textDir})
      archivo<-renderText({input$textCsv})
      rutaHadoop<-renderText({input$textHadoop})
      rutaHadoop<-rutaHadoop()

      resultado<-paste(dirr(),archivo(), sep = "")
      if (unaPasada == 0){
       unaPasada<-1
       source("primerAnalisis.R", local=TRUE)
       histogramaResultado<-hdfs.get(res_1Fraude)
       output$histoUI<-renderPlot({
         hist(histogramaResultado$val2)
       })
      }
    })
```

Listado 16. *Integración del script del primer análisis con la aplicación web.*

La línea de código sombreada es la encargada de cargar el código del script en el servidor de la aplicación.

5.3.4. Revisión del Sprint.

Al finalizar este sprint se produjo su revisión. Este sprint concluyó con la funcionalidad completa del analista de fraude. La Tabla 20 muestra la comparativa entre las horas realizadas y estimadas.

Sprint	Tareas	Horas realizadas	Horas estimadas
2	T1. Decidir que lenguaje utilizar para el script que genere las lecturas de los contadores.	3,17	3
	T2. Estudio de los tipos de fraude que se pueden cometer.	14,8	18
	T3. Creación del script que simule la generación de lecturas de contadores de agua.	-	-
	T3.1. Obtener una base de datos de inmuebles fiable.	2,5	4
	T3.2. Clasificar los inmuebles según sus características.	3	6
	T3.3. Generar los consumos de agua habiendo introducido fraude en ellos.	11	12
	T4. Implementación del script del primer análisis de fraude.	19,2	22
	T5. Implementación del script del segundo análisis de fraude.	8,8	10
	T6. Implementación del script del tercer análisis de fraude.	6,6	10
	T7. Integrar los scripts con la aplicación.	4	4
	Número total de horas:	**73,07**	**89**

Tabla 20. Trabajo realizado del Sprint 2.

En la Figura 12 se muestra de una manera más grafica la comparativa entre las horas estimadas y las horas realizadas. Para mayor claridad se van a mostrar solo las tareas, ya que las subtareas están bien definidas en la tabla anterior.

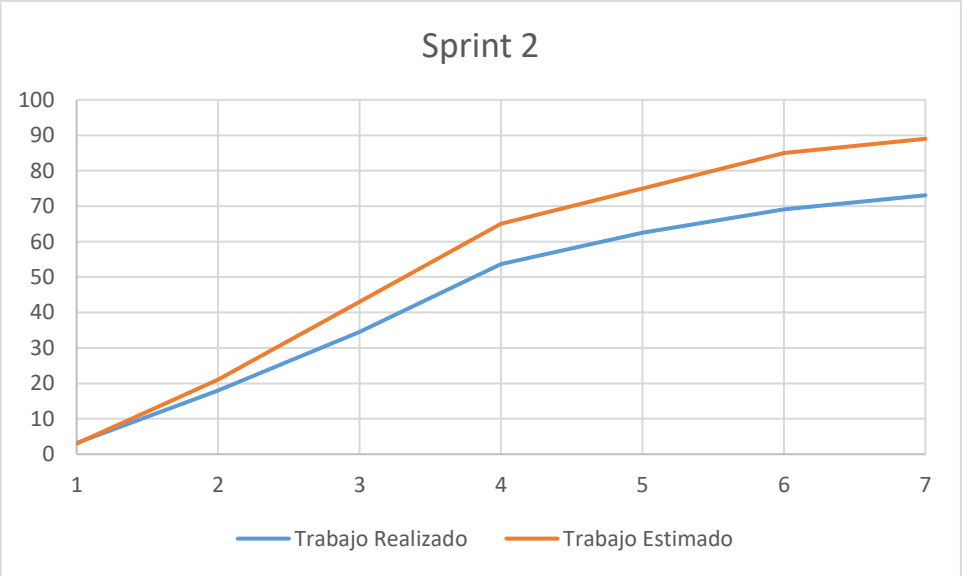

Figura 12. Gráfico Burn-Up del Sprint 2.

5.3.5. Retrospectiva del Sprint.

Para finalizar este sprint, se reunió el Equipo Scrum para revisar los objetivos marcados con la pila de producto. Este dio como resultado el prototipo del producto con la funcionalidad del analista de fraude completa.

5.4. Sprint 3.

5.4.1. Refinamiento de la pila de producto.

La pila de producto no ha sufrido modificaciones tras su revisión.

5.4.2. Planificación del Sprint.

En este caso la pila de sprint 3 está formada únicamente por la historia de usuario 4. Tabla 21.

Historia de Usuario	
Número: 4	**Usuario:** Operario
Nombre de la historia: Visualización de los análisis	
Prioridad de negocio: 90	
Esfuerzo: 34	**Sprint asignado:** 3
Programador responsable: Francisco Parreño Heredia	
Descripción: Como operario quiero obtener en la aplicación la dirección de los contadores de agua que son sospechosos de fraudes o fugas para poder personarme en la ubicación de dicho contador.	
Precondición: • Haber iniciado sesión en la aplicación con el rol de operario. • Que el analista de fraude haya iniciado los scripts para el análisis de los datos de consumo.	
Postcondición: El operario obtiene únicamente los contadores sospechosos de fraude y fugas, y la dirección de los mismos.	
Tareas (T): T1. Implementación de la funcionalidad obtener contadores.	
Artefacto o producto generado: • Funcionalidad obtención de los contadores y sus direcciones.	

Tabla 21. *Historia de Usuario 4.*

En este sprint se tuvo que añadir varias tareas adicionales que previamente no estaban contempladas en la estimación inicial. En la Tabla 22 se encuentra las tareas con su correspondiente estimación.

Sprint	Tareas	Objs*	Horas estimadas
3	T1. Implementación de operaciones que filtren únicamente los contadores sospechosos.	O.3	-
	T1.1. Filtrar los contadores del primer análisis.		6
	T1.2. Filtrar los contadores del segundo análisis.		6
	T1.3. Filtrar los contadores del tercer análisis.		6
	T2. Implementación de la funcionalidad visualizar contadores.		16
		Número total de horas:	**34**

Tabla 22. Tareas estimadas del Sprint 3.

5.4.3. Desarrollo de Tareas.

5.4.3.1. T1. Implementación operaciones que filtren los contadores sospechosos.

Para que el operario solo pueda ver los contadores sospechosos de fraude se debe filtrar los resultados de los análisis y mostrar solo los que le interesa al usuario.

La consecución de dicha tarea, se ha dividido en tres subtareas cada una dependiendo del tipo de análisis del que se van a filtrar los resultados. Los valores de filtro que se van a describir a continuación son el resultado del estudio de los fraudes en el Capítulo 3.

- *T1.1. Filtrar los contadores del primer análisis.*

  Se van a mostrar los contadores que tengan con un coeficiente de correlación de Pearson entre -1 y -0,8. El Listado 17 muestra el código para que se filtren dichos contadores.

```
observeEvent(input$PrimerFraude,{
      if (unaPasada == 0){
        unaPasada<-1
        tablaResultado<-
subset(hdfs.get(res_1Fraude),hdfs.get(res_1Fraude)$val2 < -0.8)
        output$tablaUI_1<-renderDataTable({tablaResultado}, options =
list(lengthMenu = c(5, 30, 50), pageLength = 5))
      }
    })
```

Listado 17. Código para filtrar por la correlación.

- *T1.2. Filtrar los contadores del segundo análisis.*

  Se van a mostrar los contadores cuya desviación típica sea mayor a 0,40 y el porcentaje de decremento en el consumo sea mayor del 35%. El Listado 18 muestra el código desarrollado.

```
observeEvent(input$SegundoFraude,{
      if (unaPasada == 0){
        unaPasada<-1
        tablaResultado<-
subset(hdfs.get(res_2Fraude),(hdfs.get(res_2Fraude)$val2 < 0.40 &&
hdfs.get(res_2Fraude)$val3 > 35)
        output$tablaUI_2<-renderDataTable({tablaResultado}, options =
list(lengthMenu = c(5, 30, 50), pageLength = 5))
      }
    })
```

Listado 18*. Código para filtrar según la desviación típica y el porcentaje*

- *T1.3. Filtrar los contadores del tercer análisis.*

  Se van a mostrar los contadores cuyo consumo medio anual sea menor o igual a 10 m^3. Ya que está medida según el estudio Capítulo 3 es la adecuada. El Listado 19 muestra el código desarrollado.

```
observeEvent(input$TercerFraude,{
      if (unaPasada == 0){
        unaPasada<-1
        tablaResultado<-
subset(hdfs.get(res_3Fraude),hdfs.get(res_3Fraude)$val2 < 10)
        output$tablaUI_3<-renderDataTable({tablaResultado}, options =
list(lengthMenu = c(5, 30, 50), pageLength = 5))
      }
    })
```

Listado 19*. Código para filtrar por el consumo anual.*

5.4.3.2. T2. Implementación de la funcionalidad visualizar contadores.

Para la implementación de esta tarea los scripts desarrollados anteriormente se han integrado con la aplicación para así generar en la vista de operario los resultados finales. El Listado 20 muestra las donde se van a generar estas tablas filtradas.

```
operario_main=div(

  tabItems(
    tabItem(tabName = "sospechosos",
            fluidRow(
              box(title = "Resultado Primer Análisis",status = "primary",
dataTableOutput('tablaUI_1')
              ),
              box(title = "Resultado Segundo Análisis",status = "primary",
dataTableOutput('tablaUI_2')
              ),
              box(title = "Resultado Tercer Análisis",status = "primary",
dataTableOutput('tablaUI_3')
              ),
      )
    )
  )
)
```

Listado 20*. Código de la vista del operador para generar los resultados.*

5.4.4. Revisión del Sprint.

Tras la finalización de este sprint se reunió el equipo de Equipo Scrum para revisar los objetivos con la pila del producto. Se mostró la funcionalidad completa del operario. En la Tabla 23 se muestra la comparativa entre las horas realizadas y estimadas.

Sprint	Tareas	Horas realizadas	Horas estimadas
3	T1. Implementación de operaciones que filtren únicamente los contadores sospechosos.	-	-
	T1.1. Filtrar los contadores del primer análisis.	2,5	6
	T1.2. Filtrar los contadores del segundo análisis.	2,5	6
	T1.3. Filtrar los contadores del tercer análisis.	2,5	6
	T2. Implementación de la funcionalidad visualizar contadores.	10	16
	Número total de horas:	**17,5**	**34**

Tabla 23*. Trabajo realizado del sprint 3.*

Como en los anteriores sprints en la Figura 13 se muestra la comparativa entre el trabajo estimado y el trabajo realizado.

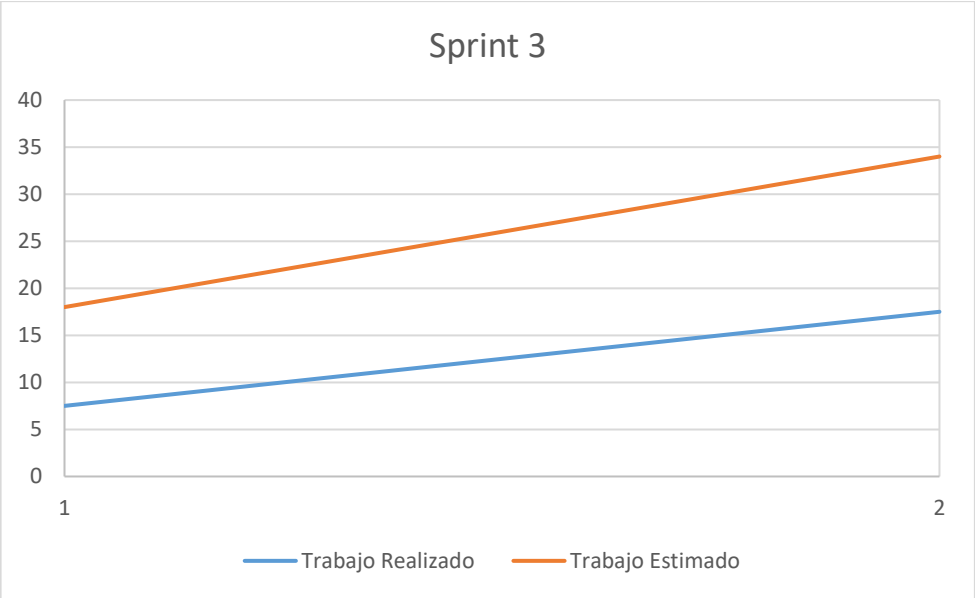

Figura 13. *Gráfico Burn-Up del Sprint 3.*

5.4.5. Retrospectiva del Sprint.

Para finalizar este sprint, se reunió el Equipo Scrum para revisar los objetivos marcados con la pila de producto. Este dio como resultado el prototipo del producto con la funcionalidad de la obtención de los contadores sospechosos en la vista del operador.

5.5. Sprint 4.

5.5.1. Refinamiento de la pila de producto.

La pila de producto no ha sufrido modificaciones tras su revisión.

5.5.2. Planificación del Sprint.

En este caso la pila de sprint 4 está formada únicamente por la historia de usuario 5. Tabla 24.

Historia de Usuario	
Número: 5	**Usuario:** Operario
Nombre de la historia: Creación documento pdf.	
Prioridad de negocio: 80	
Esfuerzo: 13	**Sprint asignado:** 4
Programador responsable: Francisco Parreño Heredia	
Descripción: Como analista de fraude y operario quiero poder obtener un informe en formato pdf de los resultados de los análisis.	
Precondición: • Haber iniciado sesión como analista de fraude u operario • Haber obtenido los resultados de los análisis de los consumos.	
Postcondición: Obtención de los documentos pdf según el rol del usuario.	
Tareas (T): T1. Generar varios pdf que tengan el contenido de los resultados de los análisis y de los contadores con sus respectivas direcciones.	
Artefacto o producto generado: • Funcionalidad para la obtención del documento pdf.	

Tabla 24*. Historia de Usuario del Sprint 4.*

La Tabla 25 muestra la estimación realizada de la tarea de esta historia de usuario.

Sprint	Tareas	Objs*	Horas estimadas
4	T1. Generar un pdf que tengan el contenido de los resultados de los análisis y de los contadores con sus respectivas direcciones.	O.4	13
		Número total de horas:	**13**

Tabla 25*. Tarea estimada del Sprint 4.*

5.5.3. Desarrollo de Tareas.

5.5.3.1. T1. Generar un pdf con los resultados de los análisis y sus direcciones.

Una vez obtenidos los resultados de los contadores sospechosos de fraude se tiene la opción de importar estos datos a un documento pdf ya que así se evita el problema de que la aplicación no funcione por cualquier situación.

El Listado 21 muestra cómo se exportaría uno de los resultados.

```
observeEvent(input$TercerFraude,{
      if (unaPasada == 0){
        unaPasada<-1
        tablaResultado<-
subset(hdfs.get(res_3Fraude),hdfs.get(res_3Fraude)$val2 < 10)
       pdf("resultados3.pdf")
        output$tablaUI_3<-renderDataTable({tablaResultado}, options =
list(lengthMenu = c(5, 30, 50), pageLength = 5))
       dev.off()
      }
    })
```

Listado 21. *Código para la generación del pdf.*

5.5.4. Revisión del Sprint.

Al finalizar el sprint se reunió el Equipo Scrum para revisar los objetivos con la pila del producto. En la Tabla 26 se muestra la comparativa entre el trabajo realizado y el trabajo estimado.

Sprint	Tareas	Horas realizadas	Horas estimadas
4	T1. Generar un pdf que tengan el contenido de los resultados de los análisis y de los contadores con sus respectivas direcciones.	10	13
	Número total de horas:	**10**	**13**

Tabla 26. *Trabajo realizado del sprint 4.*

Se ha prescindido del grafico burn-up ya que para una tarea que más claro en la tabla anterior.

5.5.5. Retrospectiva del Sprint.

Para finalizar este sprint, se reunió el Equipo Scrum para revisa los objetivos marcados con la pila de producto. Se vio que la funcionalidad de exportar los datos a pdf fue resuelta con éxito.

5.6. Sprint 5.

5.6.1. Refinamiento de la pila de producto.

La pila de producto no ha sufrido modificaciones tras su revisión.

5.6.2. Planificación del Sprint.

En este caso la pila de sprint 5 está formada únicamente por la historia de usuario 2. Esa historia de usuario se muestra en la Tabla 27.

Historia de Usuario	
Número: 2	**Usuario:** Analista de fraude
Nombre de la historia: Creación catálogo en Big Data Discovery.	
Prioridad de negocio: 75	
Esfuerzo: 21	**Sprint asignado:** 5
Programador responsable: Francisco Parreño Heredia	
Descripción: Como analista de fraude quiero poder crear desde la aplicación un catálogo de datos en la herramienta Big Data Discovery abstrayéndome de toda la lógica que eso implica.	
Precondición: Haber iniciado sesión en la aplicación teniendo el rol de analista de fraude y tener instalada la herramienta Big Data Discovery.	
Postcondición: El analista de fraude habrá creado el catálogo de datos dentro de Big Data Discovery.	
Tareas (T): T1. Implementación para la posterior creación de un proyecto en Big Data Discovery. T2. Integración del script en la aplicación.	
Artefacto o producto generado: • Funcionalidad creación del catálogo.	

Tabla 27. Historia de Usuario del Sprint 5.

En la Tabla 28 se muestra la estimación que se ha realizado de cada una de las tareas.

Sprint	Tareas	Objs*	Horas estimadas
5	T1. Implementación para la posterior creación de un proyecto en Big Data Discovery (BDD).	O.5	6
	T2. Integración del script en la aplicación.		15
		Número total de horas:	**21**

Tabla 28. Tareas estimadas del Sprint 5.

5.6.3. Desarrollo de Tareas.

5.6.3.1. T1. Implementación para la creación de un proyecto en BDD.

Después de un estudio a la documentación de la herramienta, la solución obtenida ha sido la de realizar una llamada al sistema para arrancar el script que se muestra en el Listado 22.

```
#!/bin/sh

PWD=$(dirname $0)
BDD_HADOOP_FATJAR=/u01/bdd/v1.1.1/BDD-
1.1.1.13.11/common/hadoop/lib/bddHadoopFatJar.jar
HADOOP_CONF_DIR=/u01/bdd/v1.1.1/BDD-1.1.1.13.11/common/hadoop/conf/
EDP_CLASSPATH=$PWD/libs/*:$PWD/config/:$BDD_HADOOP_FATJAR:$HADOOP_CONF_DIR

java -cp $EDP_CLASSPATH com.oracle.endeca.pdi.EdpCli edp.properties "$@"
```

Listado 22. Script data_processing_CLI

Pero la aplicación al hacer la llamada al sistema arrastraba las variables de entorno de la consola de R Studio porque lo se tuvo que clonar las variables de entorno de un terminal nativo del sistema.

En el Anexo C se encuentra el código completo.

5.6.3.2. T2. Integración con la aplicación.

Como en los sprints anteriores, la integración en la aplicación se basa mediante eventos que se llevan a cabo cuando recibe una acción. En el Listado 23 se muestra el código.

```
observeEvent(input$HiveBDD,{
     tHive<-renderText({
        input$tablaHive})
        ejecutarScript<-paste("./script_dp_CLI.sh ","tHive()",sep="" )
        system("./script_dp_CLI.sh movie")
      })
```

Listado 23. Integracion del script en el sprint 5.

5.6.4. Revisión del Sprint.

Al finalizar este Sprint se produjo su revisión. La Tabla 29 muestra la comparativa entre las horas realizadas y estimadas.

Sprint	Tareas	Horas realizadas	Horas estimadas
5	T1. Implementación para la posterior creación de un proyecto en Big Data Discovery.	4	6
	T2. Integración del script en la aplicación.	11	15
	Número total de horas:	**15**	**21**

Tabla 29. Trabajo realizado del sprint 5.

En la Figura 14 se pude observar la comparación entre el trabajo real y el trabajo estimado.

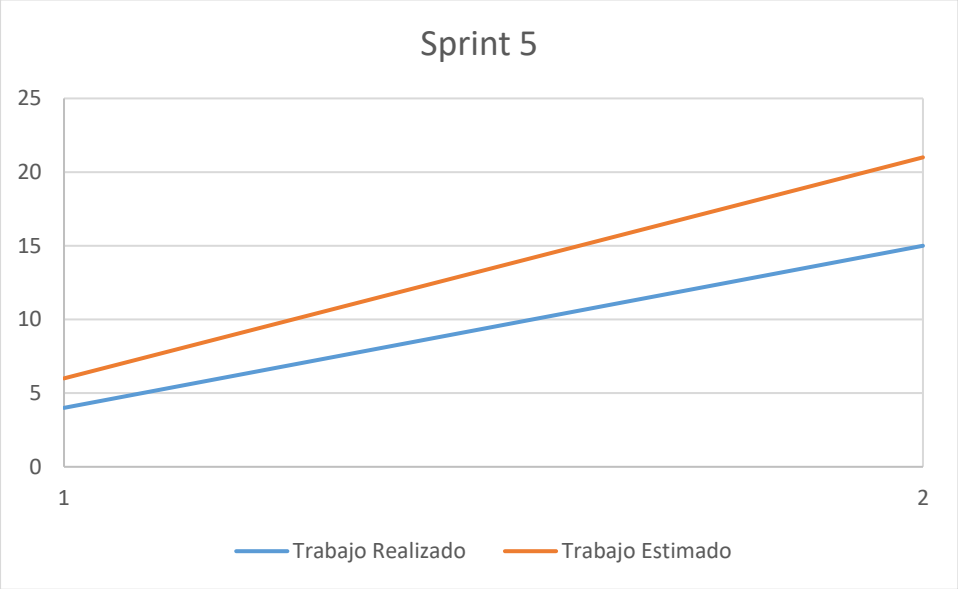

Figura 14. Gráfico Burn-Up Sprint 5.

5.6.5. Retrospectiva del Sprint.

Para finalizar este sprint, se reunió el Equipo Scrum para revisa los objetivos marcados con la pila de producto. Se vio que la opción de introducir desde la aplicación un catálogo de datos en la herramienta Big Data Discovery se había realizado con éxito.

5.7. Sprint 6.

5.7.1. Planificación del Sprint.

En este sprint se va a realizar la finalización del proyecto, tanto la parte de la documentación como la preparación de una demostración final. La Tabla 30 se muestra la estimación de cada tarea.

Sprint	Tareas	Objs*	Horas estimadas
6	T1. Creación de una demostración del sistema.	-	5
	T2. Realización de la documentación del TFG		74
	T3. Realización de un manual de usuario.		10
		Número total de horas:	**89**

Tabla 30. Tareas estimadas del Sprint 6.

5.7.2. Desarrollo de Tareas.

En este sprint se abordó las tareas de la preparación de los datos para la demostración o prueba de concepto con el propietario del producto, así como la documentación final del proyecto y su manual de usuario.

5.7.3. Revisión del Sprint.

Al finalizar este Sprint se produjo su revisión. La Tabla 31 muestra la comparativa entre las horas realizadas y estimadas.

Sprint	Tareas	Horas realizadas	Horas estimadas
6	T1. Creación de una demostración del sistema.	4,50	5
	T2. Realización de la documentación del TFG	65,30	74
	T3. Realización de un manual de usuario.	10	10
	Número total de horas:	**79,8**	**89**

Tabla 31. *Trabajo realizado del sprint 6.*

En la Figura 15 se pude observar la comparación entre el trabajo real y el trabajo estimado.

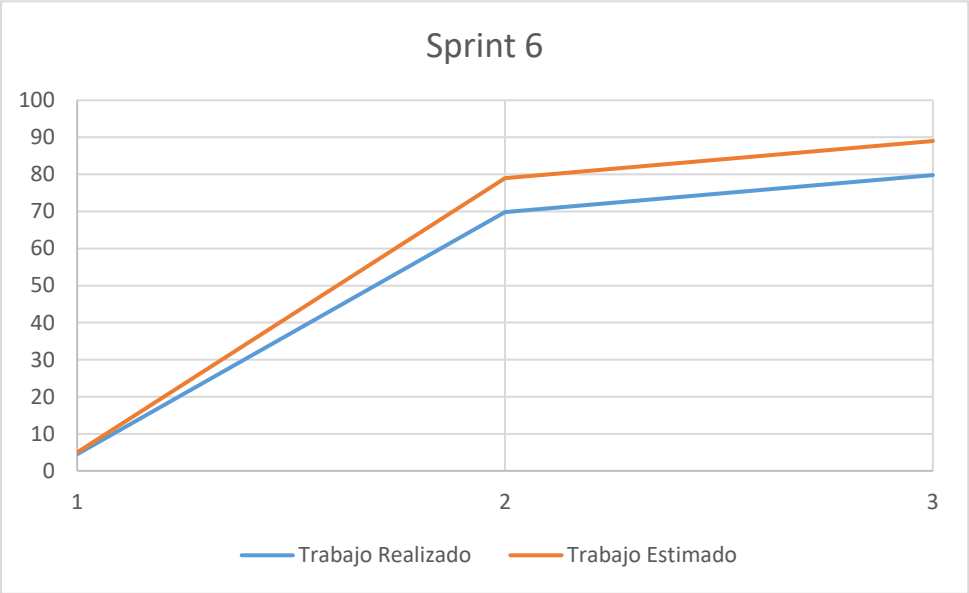

Figura 15. *Gráfico Burn-Up del Sprint 6.*

5.7.4. Retrospectiva del Sprint.

Para finalizar este sprint, se reunió el Equipo Scrum para revisar los objetivos marcados en la pila de producto. Se vio que como la finalización de dicho sprint la realización de la memoria del TFG.

CAPÍTULO 6

CONCLUSIONES Y PROPUESTAS

En este capítulo se exponen la consecución de los objetivos principal y parciales marcados durante el Capítulo 2 de este documento. Además de hacer referencia a posibles propuestas de futuro para ampliar el alcance del proyecto y una opinión personal de proyectando tras la finalización de dicha memoria.

6.1. Consecución de los objetivos del proyecto.

El objetivo principal del TFG marcado en el Capítulo 2 era el siguiente:

El objetivo principal de este TFG es la realización de un entorno Big Data en el que se realicen análisis de detección de fraudes y fugas en contadores de agua y mostrar los resultados mediante una aplicación web, para que ayuden a la eficiencia de la organización.

Este objetivo ha sido cumplido una vez ha finalizado la realización del proyecto y documentado a medida que se iban cumpliendo los objetivos parciales que se muestran en la Tabla 32.

Objetivos Parciales		Consecución	Sprint de la consecución.
1	Creación de un script que genere la simulación de las lecturas de los contadores digitales y posterior almacenamiento de datos.	✓	Sprint 2.
2	Implementación de los algoritmos de detección de fraude usando el lenguaje estadístico R	✓	Sprint 2.
3	Creación de una aplicación web que permita invocar y ejecutar los algoritmos de fraude ya implementados e interpretarlos mediante visualización de gráficas y tablas.	✓	Sprint 1. Sprint 3.
4	Generar un informe final en pdf para interpretar los resultados de los análisis.	✓	Sprint 4.
5	Configuración y creación de un catálogo datos en la herramienta Big Data Discovery, proveniente de tablas Hive mediante la aplicación web.	✓	Sprint 5.

Tabla 32. *Consecución objetivos del proyecto.*

Competencias	Justificación	Consecución
Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software	Se han desarrollado análisis de detección de fraude usando el paradigma MapReduce, además de la metodología ágil Scrum.	✓
Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.	Se ha completado la elaboración de un prototipo final que cumple con todos los requisitos marcados por el usuario.	✓
Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.	La solución final implica la integración de Hadoop y HDFS como almacenamiento de los consumos y los resultados de los análisis con la aplicación web desarrollada en Shiny como todas las conexión con la base de datos Oracle Database 12c.	✓
Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.	Todo el proyecto ha sido desarrollado y desplegado sobre la máquina virtual Big Data Lite 4.3.1. En base a los requisitos de los análisis de detección de fraude, el lenguaje principal utilizado ha sido R.	✓
Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse.	El uso de Scrum como metodología de gestión del proyecto, permitirá abordar y tratar potenciales riesgos el desarrollo, para ello se ha desarrollado en el sprint inicial un plan de riesgos.	✓
Capacidad para diseñar soluciones apropiadas en uno o más dominios de aplicación utilizando métodos de la ingeniería del software que integren aspectos éticos, sociales, legales y económicos.	El proyecto requiere una etapa inicial de identificación de los requisitos de datos para el dominio de gestión de recursos hidrológicos y para la detección de fraudes en dicho dominio. Desarrollar esta habilidad, permitirá que en un futuro se puedan aplicar a otros dominios como la salud o las utilities.	✓

Tabla 33*. Consecución de competencias.*

6.2. Posibles ampliaciones del proyecto.

Durante el desarrollo del TFG han surgido diferentes propuestas para mejorar el sistema, que debido a diversos factores como el tiempo, no han sido implementadas con la finalización del TFG, algunas de estas ideas son:

- Desarrollo del sistema en tiempo real, es decir, tener el sistema conectado al centro de datos donde llegan las lecturas y ejecutar los análisis sobre ellos.
- Identificar patrones de consumo para predecir posibles anomalías futuras con algunos contadores.
- Desplegar el sistema Big Data en múltiples nodos para mejorar el tiempo de respuesta de los análisis.

6.3. Opinión personal.

El desarrollo de este TFG ha tenido un impacto personal y profesional hacia mi persona. Profesional porque se ha realizado el TFG dentro del convenio FORTE con la empresa avanttic Consultoría Tecnológica lo cual me ha ayudado a adquirir unos conocimientos de tecnologías Oracle que difícilmente hubiesen sido alcanzados de no estar allí.

También he descubierto que el mundo Big Data es inmenso, ya que la competencia en este sector está marcando un antes y un después en la analítica de información.

El impacto personal quiero destacar la satisfacción por la realización de este proyecto que pone fin a una etapa de formación académica y da comienzo muchas más etapas a nivel profesional.

Francisco Parreño Heredia

Ciudad Real, a 30 de Agosto de 2016.

CAPÍTULO 7

BIBLIOGRAFÍA Y REFERENCIAS

[1] «El derecho humano al agua y al saneamiento | Decenio Internacional para la Acción "El agua, fuente de vida" 2005-2015». [En línea]. Disponible en: http://www.un.org/spanish/waterforlifedecade/human_right_to_water.shtml. [Accedido: 26-ago-2016].

[2] J. Quevedo, R. Pérez, J. Pascual, V. Puig, G. Cembrano, y A. Peralta, «Methodology to Detect and Isolate Water Losses in Water Hydraulic Networks: Application to Barcelona Water Network», IFAC Proc. Vol., vol. 45, n.º 20, pp. 922-927, ene. 2012.

[3] A. Gandomi y M. Haider, «Beyond the hype: Big data concepts, methods, and analytics», Int. J. Inf. Manag., vol. 35, n.º 2, pp. 137-144, abr. 2015.

[4] «Escuela Superior de Informática (UCLM) » Empresas » Programa profESIonalízate». [En línea]. Disponible en: http://webpub.esi.uclm.es/spa/paginas/empresas-profesionalizate. [Accedido: 27-ago-2016].

[5] «¿Qué es un contador de agua? - Twenergy». [En línea]. Disponible en: https://twenergy.com/a/que-es-un-contador-de-agua-1674. [Accedido: 01-sep-2016].

[6] «Big Data Analytics, 1st Edition | David Loshin | ISBN 9780124173194». [En línea]. Disponible en: http://store.elsevier.com/Big-Data-Analytics/David-Loshin/isbn-9780124173194/. [Accedido: 30-ago-2016].

[7] «What Is Big Data? - Gartner IT Glossary - Big Data». [En línea]. Disponible en: http://www.gartner.com/it-glossary/big-data/. [Accedido: 30-ago-2016].

[8] «EECS6893-BigDataAnalytics-Lecture1.key - EECS6893-BigDataAnalytics-Lecture1.pdf». [En línea]. Disponible en: http://www.ee.columbia.edu/~cylin/course/bigdata/EECS6893-BigDataAnalytics-Lecture1.pdf. [Accedido: 30-ago-2016].

[9] «MapReduce Tutorial». [En línea]. Disponible en: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html. [Accedido: 31-ago-2016].

[10] «Welcome to Apache™ Hadoop®!» [En línea]. Disponible en: http://hadoop.apache.org/. [Accedido: 30-ago-2016].

[11] «HDFS Architecture Guide». [En línea]. Disponible en: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html. [Accedido: 30-ago-2016].

[12] L. Ding, C. Pang, L. M. Kew, L. C. Jain, R. J. Howlett, I. Monedero, F. Biscarri, J. I. Guerrero, M. Roldán, y C. León, «Knowledge-Based and Intelligent Information &

Engineering Systems 19th Annual Conference, KES-2015, Singapore, September 2015 ProceedingsAn Approach to Detection of Tampering in Water Meters», Procedia Comput. Sci., vol. 60, pp. 413-421, ene. 2015.

[13] «Microsoft Word - NP_ESSA2013.doc - np934.pdf». [En línea]. Disponible en: http://www.ine.es/prensa/np934.pdf. [Accedido: 30-ago-2016].

[14] «Gestión ágil de proyectos software», 233gradosdeti.com, 14-oct-2013. .

[15] «Gestión de proyectos con Scrum Manager - sm_proyecto.pdf». [En línea]. Disponible en: http://www.scrummanager.net/files/sm_proyecto.pdf. [Accedido: 30-ago-2016].

[16] «Outlook.com, el correo electrónico personal gratuito de Microsoft». [En línea]. Disponible en: https://www.microsoft.com/es-es/outlook-com/?cb=v8ho. [Accedido: 30-ago-2016].

[17] «Programas de software de hoja de cálculo | Prueba gratuita de Excel». [En línea]. Disponible en: https://products.office.com/es-es/excel. [Accedido: 30-ago-2016].

[18] «Hatjitsu :: Online Scrum Planning Poker for Agile Projects». [En línea]. Disponible en: https://tools.wmflabs.org/hatjitsu/. [Accedido: 30-ago-2016].

[19] «Planning Poker® | Crisp - Get agile with Crisp». [En línea]. Disponible en: https://www.crisp.se/bocker-och-produkter/planning-poker. [Accedido: 31-ago-2016].

[20] «Free UML Tool». [En línea]. Disponible en: https://www.visual-paradigm.com/solution/freeumltool/?gclid=Cj0KEQjw3ZS-BRD1xu3qw8uS2s4BEiQA2bcfMyI1aNvb5HGXTunkYvALb4hYj-uWVNQ0OTqXOiEEbvMaAiN28P8HAQ. [Accedido: 30-ago-2016].

[21] «Balsamiq Mockups | Balsamiq». [En línea]. Disponible en: https://balsamiq.com/products/mockups/. [Accedido: 31-ago-2016].

[22] «GanttProject: free desktop project management app». [En línea]. Disponible en: http://www.ganttproject.biz/. [Accedido: 31-ago-2016].

[23] «Welcome to Python.org». [En línea]. Disponible en: https://www.python.org/. [Accedido: 31-ago-2016].

[24] «R: The R Project for Statistical Computing». [En línea]. Disponible en: https://www.r-project.org/. [Accedido: 31-ago-2016].

[25] «RStudio – Open source and enterprise-ready professional software for R». [En línea]. Disponible en: https://www.rstudio.com/. [Accedido: 31-ago-2016].

[26] «The Comprehensive R Archive Network». [En línea]. Disponible en: https://cran.r-project.org/. [Accedido: 31-ago-2016].

[27] «Shiny». [En línea]. Disponible en: http://shiny.rstudio.com/. [Accedido: 31-ago-2016].

[28] «Shiny Dashboard». [En línea]. Disponible en: https://rstudio.github.io/shinydashboard/. [Accedido: 31-ago-2016].

[29] «Database 12c | Oracle». [En línea]. Disponible en: https://www.oracle.com/database/index.html. [Accedido: 31-ago-2016].

[30] «Oracle SQL Developer Downloads». [En línea]. Disponible en: http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html. [Accedido: 31-ago-2016].

[31] «Microsoft Word | Software de procesamiento de texto y documentos». [En línea]. Disponible en: https://products.office.com/es-es/word. [Accedido: 31-ago-2016].

[32] «Zotero | Home». [En línea]. Disponible en: https://www.zotero.org/. [Accedido: 31-ago-2016].

[33] «GIMP - Downloads». [En línea]. Disponible en: https://www.gimp.org/downloads/. [Accedido: 31-ago-2016].

[34] «PrtScr download». [En línea]. Disponible en: http://www.fiastarta.com/PrtScr/Download.html. [Accedido: 31-ago-2016].

[35] «Dirección y Gestión de Proyectos Informáticos - buenas_practicas_proyectos_informaticos.pdf». [En línea]. Disponible en: http://www.edutecne.utn.edu.ar/proyectos_informaticos/buenas_practicas_proyectos_informaticos.pdf. [Accedido: 31-ago-2016].

ANEXO A. Manual de usuario.

En este anexo se realizara un manual de usuario para la aplicación web desarrollada en este TFG. Como se ha comentado en todo el documento, la aplicación tiene dos perfiles bien definidos: Analista de fraude y Operario.

Primero se va a mostrar la funcionalidad el Analista de fraude que una vez realice los análisis para detectar los contadores sospechosos, el operario los podrá ver junto con su correspondiente dirección en la que se encuentra el contador.

A.1. Analista de fraude.

A.1.1. Inicio de Sesión.

Para este manual se ha creado previamente un analista de fraude:

- Usuario: *analista*
- Contraseña: *analista*

La Figura 16 muestra como un usuario introduce los campos en usuario y contraseña e inicia la sesión.

Figura 16. *MU_Iniciar sesion*

A.1.2. Página Principal.

El analista se encuentra en la página principal de la aplicación en la que puede ejecutar los tres tipos de análisis. Figura 17.

Debe indicar en que directorio se encuentra los ficheros de consumo que van a ser analizados y el directorio en el que se van a almacenar en el HDFS para poder lanzar las tareas Map y Reduce.

Figura 17. MU_Página principal

A.1.3. Resultados de los análisis.

La Figura 18 muestra los resultados de la ejecución de los tres análisis de detección. Con esto, el operario debe iniciar sesión en la aplicación para recoger esos análisis filtrados según los criterios previamente desarrollados.

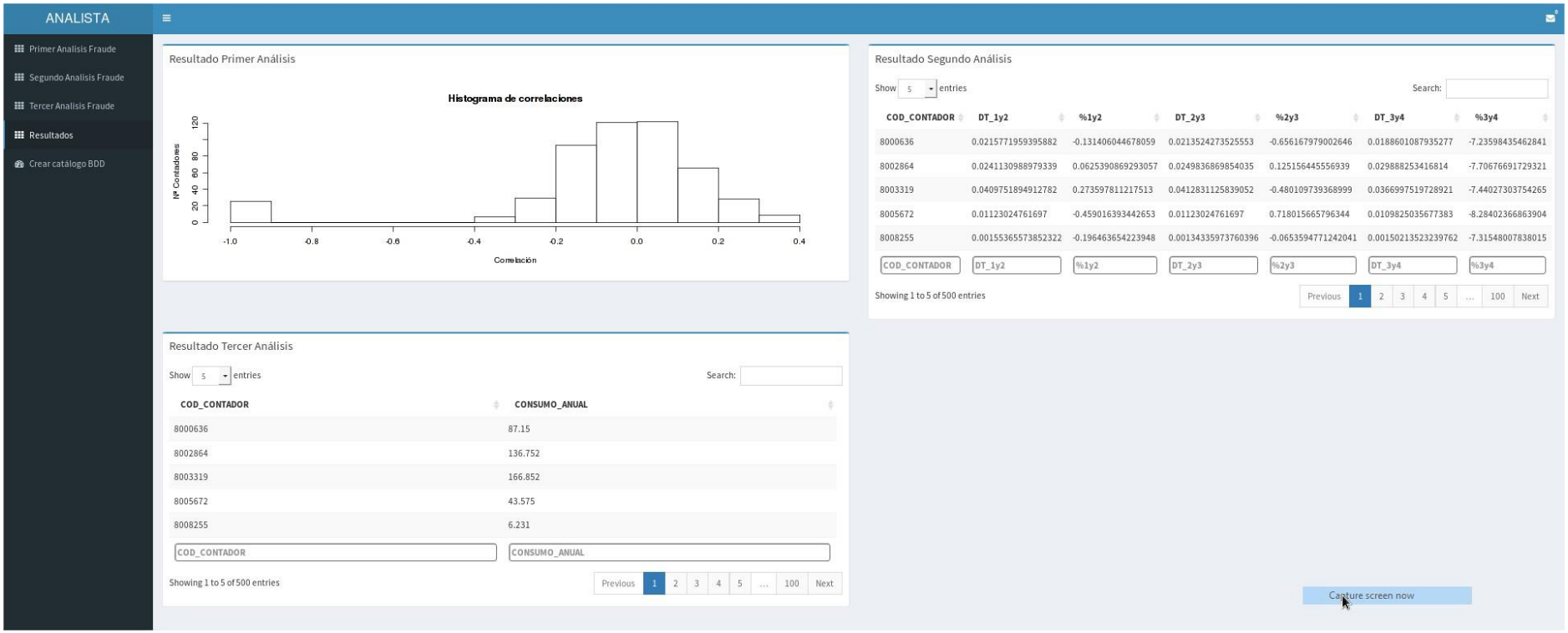

Figura 18. *MU_Resultados de los análisis. Vista analista*

A.2. Operario.

A.2.1. Inicio de Sesión.

Para este manual se ha creado previamente un operario:

- Usuario: *operario*
- Contraseña: *operario*

A.2.2. Página principal.

La Figura 19 muestra el resultado de los contadores que deben ser revisados.

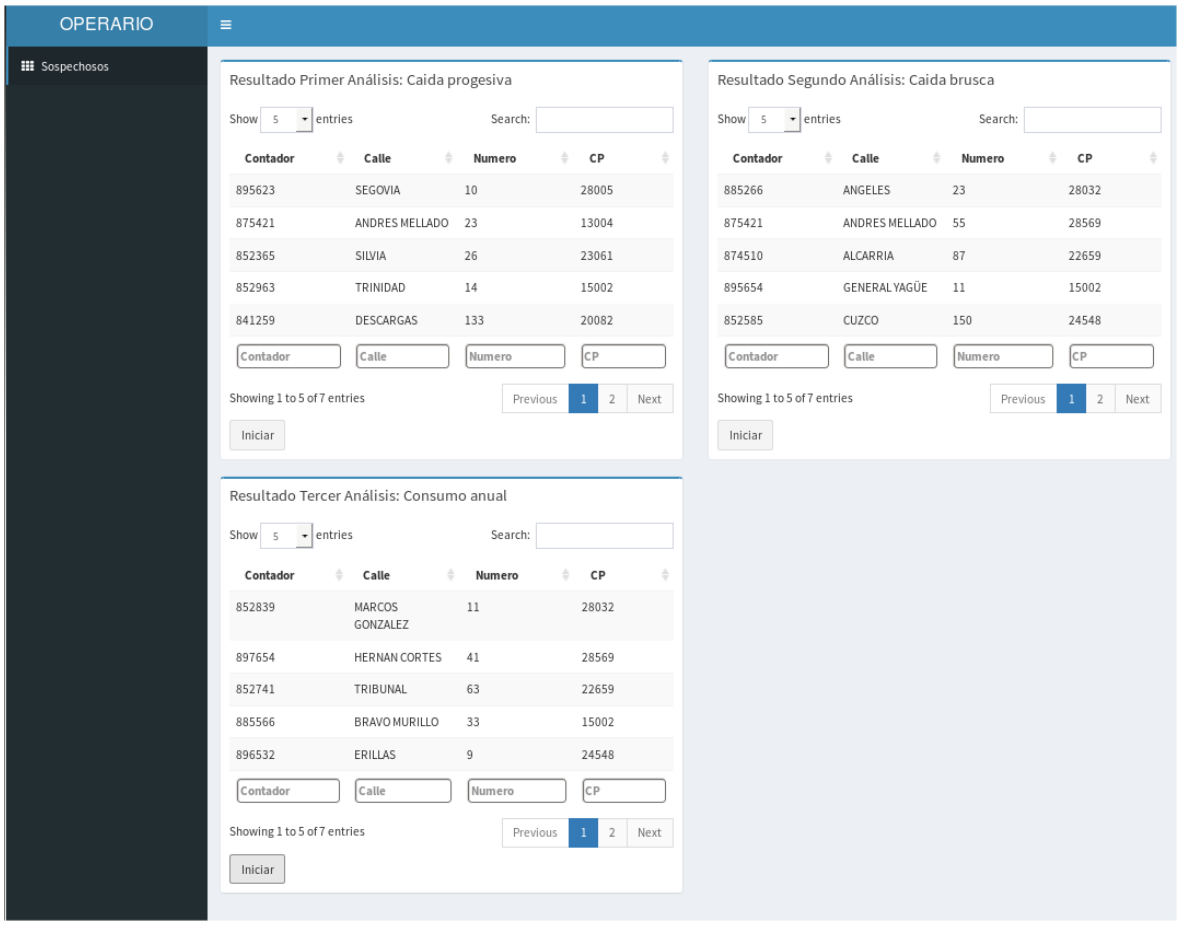

Figura 19. MU_Resultados de los analsisis. Vista Operario.

ANEXO B. Script generador de lecturas.

```
'''
@author: francisco.parreno

Este script simula el consumo de un contador de agua durante un anyo,
 en el script se puede indicar la lectura entre 1 y 30 dias

'''

import csv
import datetime
import random
#import cProfile

"""Inizializamos los contadores, tenemos 5 tipos de contadores ya que
tenemos 5 clusters"""
def inicializar_contadores():
    global CONTADOR_1, CONTADOR_2, CONTADOR_3, CONTADOR_4, CONTADOR_5

    CONTADOR_1 = round(random.uniform(0.110, 0.115),3)
    CONTADOR_2 = round(random.uniform(0.115, 0.120),3)
    CONTADOR_3 = round(random.uniform(0.120, 0.125),3)
    CONTADOR_4 = round(random.uniform(0.125, 0.130),3)
    CONTADOR_5 = round(random.uniform(0.130, 0.135),3)

    global diccionario_contador
    diccionario_contador =
{1:CONTADOR_1,2:CONTADOR_2,3:CONTADOR_3,4:CONTADOR_4,5:CONTADOR_5}
#0.13 m cubicos por habitante y dia

"""El diccionario de ficheros lo utlizamos para abrir una unica vez cada
fichero csv y no tener que abrilo de una manera secuencial por cada
vuelta que haga de cada contador"""
def diccionario_ficheros():
    global fout_ENE,
fout_FEBfout_MAR,fout_ABR,fout_MAY,fout_JUN,fout_JUL,fout_AGT,fout_SEP,fo
ut_OCT,fout_NOV,fout_DIC
    fout_ENE =
open('C:/Users/francisco.parreno/Desktop/Demostracion/ENE_2014.csv', 'a')
    fout_FEB =
open('C:/Users/francisco.parreno/Desktop/Demostracion/FEB_2014.csv', 'a')
    fout_MAR =
open('C:/Users/francisco.parreno/Desktop/Demostracion/MAR_2014.csv', 'a')
    fout_ABR =
open('C:/Users/francisco.parreno/Desktop/Demostracion/ABR_2014.csv', 'a')
    fout_MAY =
open('C:/Users/francisco.parreno/Desktop/Demostracion/MAY_2014.csv', 'a')
    fout_JUN =
open('C:/Users/francisco.parreno/Desktop/Demostracion/JUN_2014.csv', 'a')
    fout_JUL =
open('C:/Users/francisco.parreno/Desktop/Demostracion/JUL_2014.csv', 'a')
    fout_AGT =
open('C:/Users/francisco.parreno/Desktop/Demostracion/AGT_2014.csv', 'a')
    fout_SEP =
open('C:/Users/francisco.parreno/Desktop/Demostracion/SEP_2014.csv', 'a')
    fout_OCT =
open('C:/Users/francisco.parreno/Desktop/Demostracion/OCT_2014.csv', 'a')
```

```
    fout_NOV =
open('C:/Users/francisco.parreno/Desktop/Demostracion/NOV_2014.csv', 'a')
    fout_DIC =
open('C:/Users/francisco.parreno/Desktop/Demostracion/DIC_2014.csv', 'a')

    global diccionario_mes
    diccionario_mes =
{1:fout_ENE,2:fout_FEB,3:fout_MAR,4:fout_ABR,5:fout_MAY,6:fout_JUN,
7:fout_JUL,8:fout_AGT,9:fout_SEP,10:fout_OCT,11:fout_NOV,12:fout_DIC}

"""Si determinamos que el contador que toca simular su consumo no es
fraudulento, utilizamos esta funcion"""
def consumoInmueble_NO_fraude (tipoInmueble, numHab, fecha,
intervaloLec,tipoCont):
    fechaSTR = fecha[:10]
    if (numHab == 'NA'):
        numHab = 0
    else:
        numHab=int(numHab)

    if (tipoInmueble =='V'): #Viviendas
        if ((fechaSTR >= '2014-01-01') and (fechaSTR <='2014-03-31')):
#PRIMER TRIMESTRE - INVIERNO
            extras =
(diccionario_contador.get(tipoCont)*numHab*intervaloLec)
        if ((fechaSTR >= '2014-04-01') and (fechaSTR <='2014-06-30')):
#SEGUNDO TRIMESTRE - PRIMAVERA
            extras =
(diccionario_contador.get(tipoCont)*numHab*intervaloLec)
        if ((fechaSTR >= '2014-07-01') and (fechaSTR <='2014-09-30')):
#TERCER TRIMESTRE - VERANO
            extras =
(diccionario_contador.get(tipoCont)*numHab*intervaloLec)
        if ((fechaSTR >= '2014-10-01') and (fechaSTR <='2014-12-31')):
#CUARTO TRIMESTRE - OTOnO
            extras =
(diccionario_contador.get(tipoCont)*numHab*intervaloLec)

    if (tipoInmueble =='O' or tipoInmueble =='C'): #Oficinas: Gasto de
Office 0,15 l/s --- Sanitario con deposito 0,10 l/s
        extras = (diccionario_contador.get(tipoCont))

    if ((tipoInmueble =='I') or (tipoInmueble =='Z') or (tipoInmueble
=='G') or (tipoInmueble =='M') or (tipoInmueble =='J') or (tipoInmueble
=='B')):#Industrial
        extras = (diccionario_contador.get(tipoCont))

    return round(extras, 3)

"""Si determinamos que el contador que toca simular su consumo es
fraudulento, utilizamos esta funcion"""
def consumoInmueble_SI_fraude_1 (tipoInmueble, numHab,
intervaloLec,tipoCont, lectura_actual):
    extrasI = 0
    lectura_total = int(365/intervaloLec)
    if (numHab == 'NA'):
        numHab = 0
    else:
        numHab=int(numHab)

    maxima_lectura = random.uniform(0.150, 0.155)
```

```
    minima_lectura = random.uniform(0.100, 0.110)

    if (tipoInmueble =='V'): #Viviendas
        extrasI = ((maxima_lectura-((maxima_lectura-
minima_lectura)*(lectura_actual/lectura_total)) + random.uniform(-
0.010,0.010))*numHab*intervaloLec)
    else:
        extrasI = ((maxima_lectura-((maxima_lectura-
minima_lectura)*(lectura_actual/lectura_total)) + random.uniform(-
0.005,0.005))*intervaloLec)
    return round(extrasI, 3)

"""------------------------------------------------------------"""
def batch(intervaloLec):
    recuento = 1
    lectura_actual=1
    diccionario_ficheros()
    fechaIni =
datetime.datetime(2014,1,1,random.randint(19,23),random.randint(0,58),
random.randint(0,59))
    fechaFin = datetime.datetime(2014,12,31,23,59,59)
    FechaSum = datetime.timedelta(days=intervaloLec)
    f_fraude = open('contadores_fraude.csv', 'a')
    writer_fraude = csv.writer(f_fraude, lineterminator='\n')
    with open('DS_Contadores_Muestra.csv', 'r') as f_in:

        """N_CONT,NUM_HAB,CLUSTER,COD_USO --- Cabecera archivo
DS_N_CONT"""

        reader = csv.reader(f_in, lineterminator='\n')
        next(reader)
        for row in reader:
            #El indice de la lectura inicial
            variable =False
            indLecturaGENERAL = round(random.uniform(100, 523), 3)
#Truncamos el valor de la lectura para que tenga este formato XXX.XXX
            lectura_actual=1

            porcentaje = random.randint(1,100)

            while (fechaIni <= fechaFin):
                inicializar_contadores()
                writer = csv.writer(diccionario_mes.get(fechaIni.month),
lineterminator='\n')
                if (porcentaje <=5):
                    consumo = consumoInmueble_SI_fraude_1(row[3],row[1],
intervaloLec, int(row[2]), lectura_actual)
                    lectura_actual=lectura_actual+1
                    variable =True
                else:
                    consumo =
consumoInmueble_NO_fraude(row[3],row[1],str(fechaIni), intervaloLec,
int(row[2]))
                fila = (row[0], round(indLecturaGENERAL,3),
fechaIni,row[0][0],round(consumo,3))
                writer.writerow(fila)
                indLecturaGENERAL = indLecturaGENERAL+consumo
                fechaIni += FechaSum
            recuento=recuento+1
            #numContadoresFraudulentos=numContadoresFraudulentos+1
```

```
            fechaIni =
datetime.datetime(2014,1,1,random.randint(19,23),random.randint(0,58),
random.randint(0,59))

            if (variable == True):
                fil = (row[0],"Contador Fraudulento")
                writer_fraude.writerow(fil)
    f_fraude.close()
    f_in.close()

"""Esta funcion es para coger una muestra de los contadoresy utilizarla
como ejemplo"""
def muestraContadores ():
    with open ('DS_N_CONT.csv', 'r') as f_in:
        leer = csv.reader(f_in, lineterminator='\n')
        header = next(leer)
        ultimo = [row for row in leer]

    with open ('DS_Contadores_Muestra.csv', 'w') as f_out:
        muestraCont = int(input("De cuantos contadores quieres que sea la
muestra: "))
        aleatorio = random.sample(ultimo, muestraCont)
        escribir = csv.writer(f_out, lineterminator='\n')
        escribir.writerow(header)
        for i in aleatorio:
            escribir.writerow(i)
    f_in.close()
    f_out.close()

def main():
    global lectura_actual, lectura_total, nContFraud
    boo=False
    while boo==False:
        intervaloLec = int(input("Cada cuantos dias quiere que se haga la
lectura :  "))
        if ((intervaloLec >= 1) and (intervaloLec < 30)):
            batch(intervaloLec)
            boo=True
        else:
            print("ERROR: Introduzca numero correcto\n")

if __name__ == '__main__':
    #cProfile.run('main()')
    muestraContadores()
    main()
```

ANEXO C. Script para clonar el terminal.

```
unset CDH_VERSION
export
CLASSPATH=':/u01/connectors/olh/jlib/*:/usr/lib/hadoop/*:/usr/lib/hadoop/client
/*:/u01/nosql/kv-
ee/lib/kvstore.jar:.:/u01/connectors/olh/jlib/*:/usr/lib/hadoop/*:/usr/lib/hado
op/client/*:/u01/nosql/kv-ee/lib/kvstore.jar:.'
#export DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-
Y8oSdCCWEW,guid=a83b97b66a3bce5f561392fb0000020b
unset GIT_ASKPASS
export DESKTOP_SESSION=gnome
export GDMSESSION=gnome
export GDM_KEYBOARD_LAYOUT=us
export GDM_LANG=en_US.UTF-8
export GNOME_DESKTOP_SESSION_ID=this-is-deprecated
export GNOME_KEYRING_PID=8603
#export GNOME_KEYRING_SOCKET=/tmp/keyring-w4Mhfl/socket
export GTK_RC_FILES=/etc/gtk/gtkrc:/home/oracle/.gtkrc-1.2-gnome2
export
HADOOP_CLASSPATH='/u01/orahivedp/jlib/*:/u01/connectors/olh/jlib/*:/etc/hive/co
nf:/u01/connectors/osch/jlib/*:/u01/app/oracle/product/12.1.0.2/dbhome_1/jdbc/l
ib/*:/u01/nosql/kv-ee/lib/kvstore.jar'
export HISTCONTROL=ignoredups
export HIVE_AUX_JARS_PATH=/u01/nosql/kv-
ee/lib/kvclient.jar,/u01/connectors/oxh/hive/lib/apache-
xmlbeans.jar,/u01/connectors/oxh/hive/lib/orai18n-
collation.jar,/u01/connectors/oxh/hive/lib/orai18n.jar,/u01/connectors/oxh/hive
/lib/orai18n-mapping.jar,/u01/connectors/oxh/hive/lib/orai18n-
utility.jar,/u01/connectors/oxh/hive/lib/oxh-
hive.jar,/u01/connectors/oxh/hive/lib/oxh-
mapreduce.jar,/u01/connectors/oxh/hive/lib/oxquery.jar,/u01/connectors/oxh/hive
/lib/stax2-api-3.1.1.jar,/u01/connectors/oxh/hive/lib/woodstox-core-asl-
4.2.0.jar,/u01/connectors/oxh/hive/lib/xmlparserv2_sans_jaxp_services.jar,/u01/
connectors/oxh/hive/lib/xqjapi.jar,/u01/bigdatasql_config/hive_aux_jars/hive-
hcatalog-core.jar
export IMSETTINGS_INTEGRATE_DESKTOP=yes
export IMSETTINGS_MODULE=none
unset LN_S
export MAIL=/var/spool/mail/oracle
unset RMARKDOWN_MATHJAX_PATH
unset RSTUDIO
unset RSTUDIO_HTTP_REFERER
unset RSTUDIO_PANDOC
unset RSTUDIO_USER_IDENTITY
unset RS_RPOSTBACK_PATH
unset R_BROWSER
unset R_BZIPCMD
unset R_DOC_DIR
unset R_GZIPCMD
unset R_INCLUDE_DIR
unset R_LIBS_SITE
unset R_LIBS_USER
unset R_PAPERSIZE
unset R_PDFVIEWER
unset R_PLATFORM
```

```
unset R_PRINTCMD
unset R_RD4PDF
unset R_SESSION_TMPDIR
unset R_SHARE_DIR
unset R_SYSTEM_ABI
unset R_TEXI2DVICMD
unset R_UNZIPCMD
unset R_ZIPCMD
unset SED
unset SPARK_HOME
unset SPARK_JAVA_OPTS

gnome-terminal -x /u01/bdd/v1.1.1/BDD-
1.1.1.13.11/dataprocessing/edp_cli/data_processing_CLI -t $1 &
```