



Claves para escalar un software de registro y diagnóstico de ordenadores

Trabajo de Fin de Grado

Grado en Ingeniería Informática (GEI)
Especialidad de Tecnologías de la Información

Memoria

Director: David Franquesa Griso de eReuse.org

Ponent: Leandro Navarro Moldesde el DAC

Autor: Jordi Nadeu Ferran

Fecha de defensa: 25 de Enero de 2021

Resum

Aquest projecte s'enfoca en les claus per avaluar i millorar l'escalabilitat d'un software de registre i diagnòstics d'ordinadors. Aquest treball de fi de grau forma part del projecte eReuse.org, projecte que promou l'economia circular dins de l'electrònica, el desenvolupament i el manteniment de programari de codi obert per a apoderar i motivar a gent al voltant del món per a crear comunitats locals, capaces de potenciar la reutilització de dispositius electrònics.

Partim d'una eina, ja existent, anomenada Workbench dins del projecte eReuse.org. Aquest programari actualment ja captura les característiques de maquinari (com el número de sèrie, model, fabricant, etc.) dels ordinadors i els seus components, realitza proves per a obtenir un diagnòstic de l'estat de l'ordinador, i per últim també esborra les dades de manera segura en cas que sigui necessari.

La finalitat d'aquest treball és per una banda avaluar un conjunt de situacions que actualment no permet a l'eina ser escalable, després d'analitzar i investigar el funcionament i l'ús del programari Workbench, i proposar una solució per a cadascuna d'aquestes situacions. Per una altra banda crear una metodologia per calcular el potencial de millora d'una solució en una situació concreta. Posteriorment comprovar la viabilitat de les solucions proposades i el potencial de millora d'escalabilitat de cadascuna d'elles. Com a conclusions obtindrem una llista prioritzada de solucions, d'on podrem saber quina solució aporta un guany major en cada situació.

Resumen

Este proyecto se enfoca en las claves para evaluar y mejorar la escalabilidad de un software de registro y diagnóstico de ordenadores. Este trabajo de fin de grado forma parte del proyecto eReuse.org, proyecto que promueve la economía circular dentro de la electrónica, el desarrollo y el mantenimiento de software de código abierto para apoderar y motivar a gente alrededor del mundo para crear comunidades locales, capaces de potenciar la reutilización de dispositivos electrónicos.

Partimos de una herramienta, ya existente, llamada Workbench dentro del proyecto eReuse.org. Este software actualmente ya captura las características de hardware (como el número de serie, modelo, fabricante, etc.) de los ordenadores y sus componentes, realiza pruebas para obtener un diagnóstico del estado del ordenador, y por último también borra los datos de forma segura en caso que sea necesario.

La finalidad de este trabajo es por un lado definir un conjunto de situaciones que actualmente no permite a la herramienta ser escalable, después de analizar e investigar el funcionamiento y el uso del software Workbench, y proponer una solución para cada una de estas situaciones. Por otro lado crear una metodología para calcular el potencial de mejora de una solución en una situación concreta. Posteriormente comprobar la viabilidad de las soluciones propuestas y el potencial de mejora de escalabilidad de cada una de ellas. Como conclusiones obtendremos una lista priorizada de soluciones, de donde podremos saber qué solución aporta una ganancia mayor en cada situación.

Abstract

This project focuses on the keys to evaluating and improving scalability of a computer recording and diagnostic software. This bachelor's degree work is part of the eReuse.org project, which promotes the circular economy within electronics, the development and maintenance of open source software to empower and motivate people around the world to create local communities, capable of enhancing the reuse of electronic devices.

We start from an existing tool called Workbench within the eReuse.org project. This software currently already captures hardware characteristics (such as serial number, model, manufacturer, etc.) of computers and their components, performs tests to obtain a diagnosis of the computer's state, and finally also erases data securely in case it is needed.

The purpose of this work is, on the one hand, to define a set of situations that currently do not allow the tool to be scalable, after analyzing and investigating the working and use of the Workbench software, and to propose a solution for each of these situations. On the other hand, to create a methodology to calculate the improvement potential of a solution in a specific situation. Then check the feasibility of the proposed solutions and the potential for improvement of scalability of each of them. As conclusions we will obtain a prioritized list of solutions, from where we will be able to know which solution brings a greater gain in each situation.

Agradecimientos

Este camino ha sido posible gracias a todas esas personas que han creído en mí y me han apoyado en las situaciones difíciles. Especialmente a mis padres, hermana y pareja cuyo amor y soporte incondicional me ha hecho crecer como persona.

También querría agradecer a David Franquesa y a Leandro Navarro, los directores del proyecto, que me recogieron cuando era todavía un niño y me dieron la oportunidad de brillar. He aprendido muchísimo trabajando a su lado.

Por supuesto enormemente agradecido a todas esas personas involucradas en el proyecto de eReuse.org y el genial equipo que hay en Pangea.org, sin ellos tampoco hubiera sido posible realizar este proyecto. Gracias Stephan, Mireia, Manos, Lorena, Maite, Paco, Dani, y con especial mención para Xavier quien me ha demostrado que podemos aprender cada día y que todo se puede hacer con actitud positiva y ganas.

También una mención especial para Cayo y Santiago, por todo lo que me han enseñado de tecnología y por sus muchísimas contribuciones en el proyecto, realmente estoy muy agradecido de haberles tenido como apoyo durante los baches duros que han ido surgiendo durante el proyecto.

Por último gracias a todos aquellas personas que no he nombrado explícitamente pero que en algún momento de este proyecto habéis aportado vuestro granito de arena, realmente sin todos vosotros no hubiera sido posible llevar a cabo mi proyecto.

A todos vosotros, muchísimas gracias.

Índice

1. Introducción y contextualización	7
1.1. Contexto	7
1.1.1. Introducción	7
1.1.2. Actores implicados	11
1.2. Estado del arte	12
1.2.1. Obtener información acerca del hardware del ordenador	12
1.2.2. Benchmarking para medir el rendimiento del ordenador	13
1.2.3. Pruebas para diagnosticar el estado funcional del ordenador	14
1.2.4. Borrado de datos para eliminar información de los discos duros	15
1.2.5. Conclusiones	15
2. Alcance del proyecto	17
2.1. Objetivos	17
2.2. Posibles obstáculos	17
2.3. Metodología	18
3. Planificación temporal	21
3.1. Descripción de tareas	21
3.1.1. Familiarización con el ecosistema eReuse	21
3.1.2. Creación de una metodología para valorar la escalabilidad	21
3.1.3. Evaluación de las situaciones planteadas por el equipo de eReuse	22
3.1.4. Definición de una posible solución sobre cada situación evaluada	22
3.1.5. Comprobación de la viabilidad de las soluciones propuestas	22
3.1.6. Aplicación de la metodología sobre las soluciones válidas	22
3.1.7. Creación de un prototipo de la solución escogida como prioritaria	23
3.1.8. Documentación y preparación de la defensa	23
3.2. Estimaciones	23
3.3. Diagrama de Gantt	24
3.4. Recursos usados	24
3.4.1. Recursos técnicos	25
3.4.2. Recursos humanos	25
4. Gestión económica	26
4.1. Presupuesto	26
4.1.1. Recursos Software	26
4.1.2. Recursos Hardware	26
4.1.3. Recursos humanos	27
4.1.4. Contingencias	27
4.1.5. Presupuesto final	27
4.1.6. Control de gestión	28
5. Evaluación de las situaciones planteadas por el equipo de eReuse	29
6. Posibles soluciones sobre la escalabilidad del Workbench	32
6.1. Solución A: Distribuir imagen ISO del Workbench por red usando un servidor PXE	32
6.1.1. Descripción de la solución	32

6.1.2.	Viabilidad de la solución	32
6.1.3.	Aplicación de la metodología	33
6.1.4.	Resultados	34
6.2.	Solución B: Sistema de depuración para la mejora continua de la herramienta	34
6.2.1.	Descripción de la solución	34
6.2.2.	Viabilidad de la solución	34
6.2.3.	Aplicación de la metodología	34
6.2.4.	Resultados	36
6.3.	Solución C: Modificar la imagen ISO genérica para cada usuario específico	36
6.3.1.	Descripción de la solución	36
6.3.2.	Viabilidad de la solución	37
6.3.3.	Aplicación de la metodología	37
6.3.4.	Resultados	38
6.4.	Solución D: Mostrar información sobre el error al usuario de forma categorizada y comprensible	38
6.4.1.	Descripción de la solución	38
6.4.2.	Viabilidad de la solución	39
6.4.3.	Aplicación de la metodología	39
6.4.4.	Resultados	40
6.5.	Solución E: Crear ordenadores virtualizados para aumentar la compatibilidad con hardware diferente	40
6.5.1.	Descripción de la solución	40
6.5.2.	Viabilidad de la solución	41
6.5.3.	Aplicación de la metodología	41
6.5.4.	Resultados	42
7.	Análisis de la escalabilidad de las soluciones validadas	43
7.1.	Lista priorizada de las soluciones	44
8.	Prototipo de la solución escogida	46
8.1.	Descripción del prototipo funcional	46
8.2.	Resultados usando el prototipo funcional	47
8.3.	Futura integración del prototipo	49
9.	Sostenibilidad y compromiso social	50
9.1.	Autoevaluación	50
9.1.1.	Dimensión ambiental	50
9.1.2.	Dimensión económica	51
9.1.3.	Dimensión social	51
10.	Conclusiones y futuro trabajo	52
10.1.	Objetivos cumplidos	52
10.2.	Conclusiones personales	52
10.3.	Trabajo futuro	53
Bibliografía		54

Índice de figuras

1.	<i>Ciclo de vida de un dispositivo electrónico</i>	8
2.	<i>Esquema del sistema Devicehub</i>	10
3.	<i>Tabla ejemplo - Mejora escalabilidad</i>	20
4.	<i>Tabla de estimaciones de las tareas</i>	23
5.	<i>Diagrama de Gantt</i>	24
6.	<i>Tabla con el presupuesto final del proyecto</i>	28
7.	<i>Tabla Metodología (Situación A - Solución A)</i>	33
8.	<i>Tabla Metodología (Situación B - Solución B)</i>	35
9.	<i>Tabla Metodología (Situación C - Solución C)</i>	38
10.	<i>Tabla Metodología (Situación D - Solución D)</i>	40
11.	<i>Tabla Metodología (Situación E - Solución E)</i>	42
12.	<i>Tabla Metodología Soluciones</i>	43
13.	<i>Lista priorizada de las soluciones</i>	44

1. Introducción y contextualización

1.1. Contexto

1.1.1. Introducción

Este proyecto es un Trabajo de Fin de Grado en Ingeniería Informática de la Facultad de Informática de Barcelona (FIB) [1] de la Universitat Politècnica de Catalunya (UPC) [2], dentro de la especialidad de Tecnologías de la Información (TI). Se lleva a cabo durante las prácticas curriculares en la empresa Pangea.org [3] dentro del proyecto eReuse.org [4].

La razón de este proyecto surge de la necesidad de mantener y mejorar las herramientas de código abierto que actualmente tiene eReuse.org. Más concretamente, la herramienta llamada Workbench [5], que captura las características de hardware (como el número de serie, modelo, fabricante, etc.) de un conjunto de ordenadores y de sus componentes. También realiza un conjunto de pruebas sobre los componentes para obtener un diagnóstico del estado actual del ordenador, y además puede realizar un borrado de los datos del disco duro de forma segura en caso que sea necesario.

Para empezar a poner un poco de contexto, lo primero que vamos a explicar son las siguientes cuestiones: qué es eReuse.org, qué valores promueve tanto dentro de la economía circular como de la comunidad de código libre, y cuál es su misión actual.

Antes de entrar directamente en las cuestiones planteadas en el párrafo anterior, vamos a definir un par de conceptos muy importantes, en los cuales se mueve el proyecto eReuse.org. Estos son la economía circular y el software de código libre o código abierto.

La economía circular [6][7] es un modelo económico mundial que tiene por objeto desvincular el crecimiento económico y el desarrollo del consumo de recursos finitos. Está enfocado a la restauración por diseño, y tiene como objetivo mantener los productos, componentes y materiales en su máxima utilidad y valor, en todo momento. A diferencia de una economía lineal, se trata de optimizar los sistemas en lugar de los componentes. Esto incluye la gestión cuidadosa de los materiales que fluyen en los ciclos biológicos y técnicos.

El software de código abierto (en inglés OSS, Open Source Software) [8] es un modelo de desarrollo de software informático en el cual el código fuente se libera bajo una licencia en la que el titular de los derechos de autor concede a los usuarios los derechos de uso, estudio, cambio y distribución del software a cualquier persona y para cualquier propósito [9]. Dentro de las licencias más usadas por la comunidad de código abierto hay la GPL (General Public License) [10], más concretamente la GPLv2 y la GPLv3. Un software es libre si otorga a los usuarios de manera adecuada las denominadas cuatro libertades: libertad de usar, estudiar, distribuir y mejorar, de lo contrario no se trata de software libre [11]. El software de código abierto puede desarrollarse de forma pública en colaboración. El software de código abierto es un ejemplo destacado de colaboración abierta [12].

El proyecto eReuse.org promueve un conjunto de herramientas de código abierto, juntamente con modelos y servicios abiertos para apoyar el alcance de la economía circular dentro de la electrónica para aumentar la vida útil y asegurando la trazabilidad hasta el reciclaje final de los dispositivos digitales. Tenemos la premisa de que muchos dispositivos digitales de empresas y organizaciones públicas son desmantelados y reciclados cuando se amortizan o están fuera de garantía, a pesar de

Cada vez hay una necesidad mayor de reutilizar en vez de reciclar prematuramente los ordenadores. Primero debemos ser conscientes de que manufacturar dispositivos digitales, como por ejemplo un ordenador, consume una gran cantidad de recursos finitos y algunos de ellos bastante difíciles y costos de obtener. Además, en un periodo de tiempo demasiado corto, se envían a reciclaje prematuramente en vez de ser reparados, restaurados o mejorados para darles otra vida útil. En el mejor de los casos, serán reciclados correctamente en uno de los lugares preparados específicamente para este cometido. En cambio, ocurre que una gran cantidad serán reciclados de manera informal en el mundo subdesarrollado o terminarán en vertederos, amenazando a las personas locales y al medio ambiente en general. Hay diversos factores que contribuyen a ello, como la falta de información o concienciación de la población en general, así como software o herramientas que faciliten la reutilización electrónica, y también estándares enfocados a la reutilización de todos estos dispositivos que cada día son más importantes para nuestras actividades cotidianas y laborales.

Por lo tanto, para que la reutilización tenga éxito, es necesario que los servicios y las tecnologías ganen eficiencia, accesibilidad y facilidad de uso. Entonces, la pregunta de investigación de este proyecto es: ¿Cómo puede la herramienta Workbench ser más eficiente, segura, fácil de usar y accesible al máximo de personas y tipos de ordenadores?

Este proyecto propone como aumentar la escalabilidad de este software libre de registro y diagnóstico de ordenadores. Para ello partimos un conjunto de herramientas de código abierto, que se detallan más adelante, y las cuales vamos a investigar y usar para definir un conjunto de cambios/puntos/mejoras clave que tienen como objetivo principal aumentar la escalabilidad del Workbench, tanto para el usuario final que usa la herramienta como para los desarrolladores que quieran mejorarla siguiendo la filosofía de código abierto. Después se evaluarán todas las soluciones propuestas anteriormente para escoger una de las soluciones como prioritaria y en la medida de lo posible realizar una aproximación de como se integraría, en un futuro, la solución dentro del ecosistema de eReuse.

Como bien expresa el título de este Trabajo Final de Grado, la finalidad del proyecto es la de obtener un sistema escalable y fácilmente utilizable para el software Workbench [5], principalmente todas las soluciones propuestas tienen que aumentar de algún modo la escalabilidad, dependiendo del contexto en el cual se use la herramienta veremos que puede variar bastante la forma de cómo hacerlo. Por ejemplo, en caso de un volumen grande de ordenadores a registrar, una forma sería reducir el tiempo de registro y diagnóstico. En cambio en un contexto de registro sin conexión a Internet la prioridad sería guardar los datos del registro en local para no tener que realizar una segunda ejecución del Workbench para poder subir la información del registro al Devicehub.

Como hay todo un conjunto de herramientas de código abierto que van a ser importantes para el proyecto, es muy recomendable antes de ponernos en materia, empezar detallando la situación actual del software que vamos a mejorar en este caso el Workbench, y también del ecosistema que ya existe dentro del proyecto eReuse.

Dentro del ecosistema de eReuse se encuentran dos grandes sistemas. Por un lado, un sistema de gestión de dispositivos, dicho de otra forma un inventario, ofrecido como una aplicación web y centrado en la gestión eficiente del ciclo de vida circular de los dispositivos, desde la reutilización hasta el reciclaje. Esta herramienta de gestión de dispositivos la vamos a llamar Devicehub [14] [15]. Es una aplicación web RESTful desarrollada en Python 3 y construida sobre el microframework Teal [16], un microframework de código abierto que también forma parte del ecosistema eReuse. Teal está basado y construido sobre Flask. Para guardar toda la información asociada al inventario Devicehub se utiliza PostgreSQL como base de datos. Tiene también una aplicación web construida con Javascript, CSS y HTML usando

el framework Angular 1. El esquema de la Figura 2 ilustra el funcionamiento general de Devicehub.

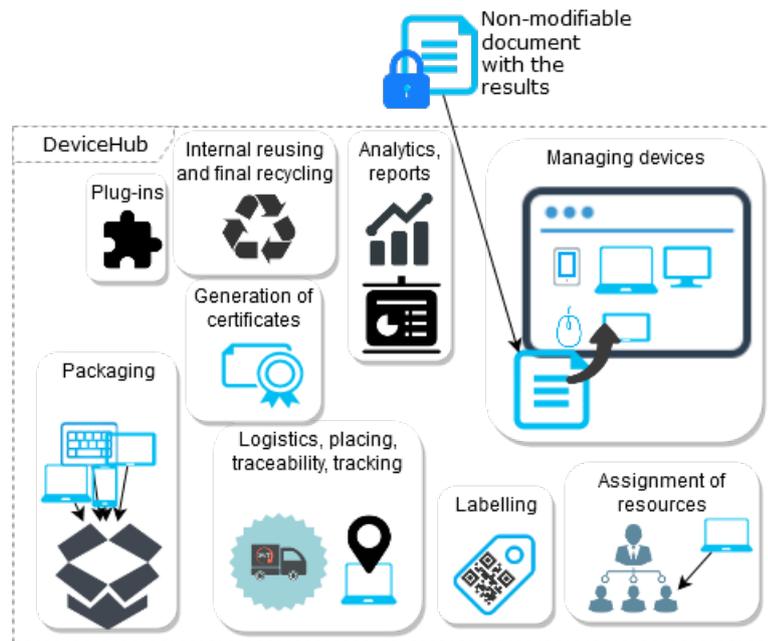


Figura 2: Esquema del sistema Devicehub

Por otro lado, está el sistema ya mencionado anteriormente, Workbench, que captura las características de hardware de los ordenadores y sus componentes, también realiza un diagnóstico del estado del ordenador, y borra los datos de forma segura en caso que sea necesario. Una de las partes más relevantes del Workbench, para este trabajo, es poder identificar de forma única cada dispositivo, para eso la herramienta necesita obtener unos datos concretos que se consideran de prioridad alta, concretamente el tipo, el fabricante, el modelo y el número de serie del dispositivo. Si obtenemos estos cuatro datos, el dispositivo puede ser identificado de forma única y por lo tanto se convierte en un dispositivo compatible para el Workbench. El Workbench es una aplicación desarrollada en Python 3, por el momento testada sobre sistemas Linux basados en Debian. Actualmente se usa como una live ISO gracias a la herramienta de código abierto que también forma parte del ecosistema eReuse; Workbench Live [17] que genera una imagen ISO basada en un sistema operativo base Debian 9 que autoejecuta la herramienta de registro y diagnóstico automáticamente al arrancar la imagen ISO. La herramienta Workbench Live está basada en un paquete de Debian llamado *live-build* [18].

Hay más piezas dentro del ecosistema de eReuse.org, pero para este proyecto no serán relevantes y para no alargar tanto la introducción no los vamos a detallar, aunque alguno puede aparecer puntualmente más adelante.

Si vamos a definir algunos conceptos necesarios para entender mejor este trabajo. Empezando por el significado de la palabra escalabilidad, según el diccionario de Cambridge [19], es un anglicismo que describe la capacidad de un negocio o sistema de crecer en magnitud. En general, también se podría definir la escalabilidad como la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para reaccionar y adaptarse sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida y sin aumentar los costes, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

El concepto de escalabilidad dentro de un software de registro y diagnóstico de ordenadores, como podría ser el Workbench, se podría definir como la capacidad de adaptación y respuesta del software con respecto al rendimiento del mismo a medida que aumentan de forma significativa la magnitud de las dimensiones, descritas en 2.3 sobre las que se quiere trabajar. Por ejemplo podría ser el número de usuarios nuevos usando el software, o el volumen de ordenadores a registrar, o la diversidad de modelos de ordenadores a registrar. Todas estas dimensiones serán definidas más adelante dentro de la metodología 2.3, concretamente en este apartado 2.3.

Después el concepto de situación, explicado en la sección 5, nos servirán de contexto y como punto de partida para la aplicación de la metodología creada y así enfocar la valoración de mejora de la escalabilidad de una forma más realista y asegurando que el conjunto de soluciones que se plantean son para solventar o mejorar situaciones reales con las que nos podemos encontrar con la herramienta Workbench dentro del proyecto eResue.org.

Por último también vamos a incorporar el concepto de madurez tecnológica, existen unos niveles de madurez tecnológica, TRLs [20] por sus siglas en inglés (*Technology Readiness Level*), definidos como un método para estimar la madurez de las tecnologías durante la concepción de un nuevo programa o solución. En este trabajo vamos a usar la escala definida por la Unión Europea y que posteriormente fue formalizada aún más mediante el estándar ISO 16290: 2013. Esta escala consiste en nueve niveles descritos a continuación:

- TRL 1 – Principios básicos observados
- TRL 2 – Concepto de tecnología formulado
- TRL 3 – Prueba experimental de concepto
- TRL 4 – Tecnología validada en laboratorio
- TRL 5 – Tecnología validada en un entorno relevante
- TRL 6 – Tecnología demostrada en un entorno relevante
- TRL 7 – Demostración del prototipo del sistema en un entorno operativo
- TRL 8 – Sistema completo y calificado
- TRL 9 – Sistema real probado en el entorno operativo

1.1.2. Actores implicados

El proyecto va dirigido a cualquier persona que quiera usar una herramienta para realizar un registro y/o diagnóstico de uno o varios ordenadores usando el software Workbench de eReuse. También puede estar enfocado a todos aquellos usuarios con conocimientos técnicos que quieran realizar mejoras o adaptar la herramienta a otros entornos, ya que se trata de un software de código abierto. Dentro de los actores implicados encontramos principalmente al:

- **Desarrollador principal:** El autor de este proyecto será la única persona que desarrolle el proyecto, así como el encargado de cumplir con todas los hitos definidos, documentar el proyecto y finalmente también presentarlo.
- **Director del proyecto:** El director de este proyecto es David Franquesa Griso, su papel principal es supervisar el cumplimiento de los objetivos definidos y el calendario establecido para el desarrollo de este proyecto. También en caso de necesidad, dar los recursos necesarios para poder proceder con el desarrollo del proyecto.

- **Supervisor del TFG:** En este caso es Leandro Navarro Moldes. Encargado de supervisar a nivel técnico y académico el cumplimiento de los requisitos necesarios dentro del proyecto, así como resolver posibles dudas durante el transcurso del proyecto y también corregir el enfoque del proyecto de forma dinámica en caso necesario.
- **Equipo de eReuse:** Son el equipo encargado de suministrar las tecnologías requeridas, gestionar y resolver posibles dudas técnicas dentro del funcionamiento de las herramientas de eReuse. También se encargan de realizar el aprovisionamiento de la infraestructura requerida para el desarrollo del proyecto.
- **Colaboradores externos:** Son personas con conocimiento técnico que realizan aportaciones al código abierto y que para nuestro proyecto podemos reutilizar para algunas de las soluciones que se plantean.
- **Usuarios finales:** Al ser un proyecto de código abierto y gratuito, cualquier usuario podrá hacer uso de las mejoras realizadas en la herramienta Workbench.

1.2. Estado del arte

Dentro del estado del arte vamos a realizar un repaso del mercado actual para ver qué hay realmente hoy en día similar al software Workbench. Como hay muchas herramientas similares, vamos a exponer solo las más populares y que se acerquen más al propósito general del Workbench. Partimos del contexto que todos los programas que se citan más adelante tienen que correr sobre un sistema operativo específico, aunque nuestro proyecto siempre vamos a trabajar con Linux y más concretamente sobre el sistema operativo Debian. También añadir que siempre se prioriza si existe alguna alternativa de código abierto antes de códigos privativos con licencias comerciales.

Como mención especial el software más similar que hemos encontrado al Workbench actualmente, aunque se ejecuta sobre Windows y no usa las mismas tecnologías, sí engloba las cuatro funciones claves explicadas más adelante. Se trata de un software privativo llamado **Aiken Workbench**, propiedad de Aiken Software LTD. Puede obtener la información del hardware, realizar benchmarking, tests y también borrado de datos. Más información es su web <https://www.aikensoftware.com/>.

Como hay tantas alternativas específicas dentro del mercado de productos similares al Workbench, vamos a clasificar dentro de las cuatro funciones claves que son; Obtener información acerca del hardware del ordenador 1.2.1, benchmarking para comprobar el rendimiento 1.2.2, pruebas o tests para diagnosticar el estado funcional 1.2.3 o borrado de datos 1.2.4.

1.2.1. Obtener información acerca del hardware del ordenador

Función principal que básicamente consiste en extraer toda la información posible del ordenador, incluidos sus componentes, tanto para identificar como también poder realizar un registro. Actualmente el Workbench usa dos paquetes diferentes para conseguir la información, el lshw [21] y el hwinfo [22]. Los dos paquetes son de código abierto bajo la licencia GPL y trabajan sobre sistemas Linux. Si se quiere profundizar más y consultar el código fuente aquí el repositorio oficial del lshw [23] y también el hwinfo [24]. Dentro del mercado actual podemos observar varias alternativas como:

- **Open Hardware Monitor.** Alternativa de código abierto disponible tanto para Linux como para Windows. Más información en su página web <https://openhardwaremonitor.org/>. También se puede consultar su código fuente en Github <https://github.com/openhardwaremonitor/openhardwaremonitor>.
- **CPU-Z.** Una de las aplicaciones más populares y usadas en Windows para obtener información sobre el hardware de una forma simple y medianamente resumida. También disponible para sistemas Android. Aunque su uso es gratuito se encuentra bajo una licencia privada, lo que no permite consultar su código fuente. Para más información visitar su web oficial <https://www.cpuid.com/softwares/cpu-z.html>.
- **I-Nex.** Es una aplicación gratuita desarrollada en Polonia y disponible solo para Linux que recoge la información de los componentes de hardware disponibles en una máquina y la muestra utilizando una interfaz de usuario similar a la popular herramienta de Windows CPU-Z. Más información en su web <https://i-nex.linux.pl/> y su código fuente está disponible en <https://github.com/i-nex/I-Nex>.
- **HWinfo.** Una solución muy similar a la CPU-Z pero mostrando la información de forma mucho más detallada. Solamente está disponible para Windows y DOS. Más información en su web <https://www.hwinfo.com/>.
- **Speccy.** Otra alternativa muy similar a CPU-Z pero solamente disponible para Windows. Hecha por los mismos desarrolladores de CCleaner, más información en su página web <https://www.ccleaner.com/speccy>.
- Hay muchos más programas similares a los mencionados anteriormente, o incluso más específicos como **GPU-Z** un programa diseñado para proporcionar información específica sobre tarjetas de vídeo.

1.2.2. Benchmarking para medir el rendimiento del ordenador

Actualmente se realizan benchmarking sobre los componentes de la CPU, la RAM y también el disco duro. Tanto la CPU como la RAM se usa una herramienta de software libre, bajo la licencia GPLv2, disponible tanto para Linux, Windows como MacOS llamada Sysbench [25]. Con el código publicado en Github [26]. En el caso del disco duro, para saber su rendimiento se usa también una herramienta libre bastante popular dentro de los sistemas Unix llamada dd [27]. Forma parte del GNU core utilities [28] que está bajo la licencia GPLv3.

Referente a las alternativas actuales sobre programas que realizan benchmarking sobre un ordenador tenemos:

- **Test Suite Phoronix.** Aplicación de software libre, bajo licencia GNU GPLv3, más popular sobre benchmarking tanto para la CPU, RAM como el disco duro. Es multiplataforma, usa la tecnología PHP y se puede descargar, consultar y mejorar el código fuente [29]. Más información en su web <https://www.phoronix-test-suite.com/>.
- **CrystalDiskmark.** Programa para saber el rendimiento del disco duro usando benchmarking. Es software libre bajo licencia MIT [30] solo disponible para Windows. Más información en su web <https://crystalmark.info/en/software/crystaldiskmark/>.

- **Geekbench 5.** Software privativo bastante usado, ya que está disponible para las cinco plataformas más usadas: Windows, Linux, MacOS, Android y iOS. Más información en su web <https://www.geekbench.com/index.html>.
- **Novabench.** Otro software privativo que se puede usar de forma gratuita pero limitada, que también es multiplataforma (Windows, Linux y MacOS). Web <https://novabench.com/>.

1.2.3. Pruebas para diagnosticar el estado funcional del ordenador

Actualmente la herramienta Workbench ejecuta dos tipos de pruebas. Una prueba de estrés para asegurar que el ordenador aguanta ciertos minutos trabajando a la máxima capacidad de procesamiento. Usa una herramienta de código abierto bajo licencia GPL llamada stress [31]. Y también un SMART test que está enfocado a diagnosticar el estado del disco duro, como resultado de esta prueba se obtiene un dato muy valioso, el tiempo de vida del disco duro. Para realizar esta prueba el Workbench usa otra herramienta de código libre bajo GPL llamada smartmontool [32].

Referente a implementaciones concretas del test SMART del disco duro o una prueba de estrés de la CPU.

- **GNOME Disks.** Alternativa específica para realizar tests y benchmarking sobre el disco duro. Se encuentra disponible para Linux dentro de la GNOME Core Applications que forma parte del entorno de escritorio estándar libre y de código abierto de GNOME. Es un programa de código libre bajo licencia GPL. Podemos obtener más información en su web oficial <https://wiki.gnome.org/Apps/Disks> y revisar su código fuente en <https://gitlab.gnome.org/GNOME/gnome-disk-utility>.
- **GSmartControl.** Realiza un diagnóstico del estado del disco duro. Programa de software libre bajo licencia GPL y disponible tanto para Windows, Linux como MacOS. La parte negativa es que no está actualizado desde 2018 y no incorpora las últimas novedades del SMART test. Más información en su web <https://gsmartcontrol.sourceforge.io/home/index.php>.
- **CrystalDiskInfo.** Programa para saber el estado del disco duro usando SMART test. Es software libre bajo licencia MIT [30] solo disponible para Windows. Más información en su web <https://crystalmark.info/en/software/crystaldiskinfo/>.
- **Memtest 86+.** Alternativa específica para realizar tests sobre la memoria RAM. Se trata de un programa libre bajo licencia GPL, y se carga fácilmente en una unidad USB de arranque a través de un instalador automático. Más información en su web <https://www.memtest.org/>.
- **HWMonitor.** Pertenece a CPUID igual que el anterior programa CPU-Z, pero este sirve para monitorizar y diagnosticar el estado del ordenador y sus componentes. Más información en su web <https://www.cpubid.com/software/hwmonitor.html>.
- **CPU-Z.** Aunque ya ha salido en el apartado 1.2.1 también se puede usar este software privativo para realizar una prueba de estrés.

1.2.4. Borrado de datos para eliminar información de los discos duros

Actualmente el Workbench permite realizar el borrado usando dos herramientas distintas, ambas de código abierto y bajo la licencia GPL. La primera se llamaba shred [33] y se usa para un borrado rápido pero menos seguro. Forma parte de la GNU core utilities. La segunda es badblocks [34] la cual realiza un borrado mucho más seguro, pero a la vez requiere más tiempo de borrado. Se encuentra dentro del conjunto de utilidades e2fsprogs [35].

Dentro del apartado de programas para realizar un borrado de datos aunque hay muchas herramientas pequeñas y específicas, a nivel global y con un uso popular podemos citar las siguientes alternativas:

- **GParted.** Herramienta de código abierto bajo licencia GPL para gestionar particiones de los discos duros y también borrar particiones. Muy usada dentro de los sistemas Linux y más concretamente dentro del entorno de escritorio estándar libre y de código abierto de GNOME. Podemos obtener más información en su web oficial <https://gparted.org/> y revisar su código fuente en <https://gitlab.gnome.org/GNOME/gparted/>.
- **Blancco Drive Eraser.** Uno de los software más reconocidos y extendidos para realizar borrados seguros de datos. Es un software privativo con un gran abanico de funciones, actualizaciones y subprogramas más específicos para la eliminación de datos. Más información en su web <https://www.blancco.com/products/drive-eraser/>.
- **DBAN.** Alternativa de código libre bajo licencia GPL para realizar borrado de datos, actualmente en propiedad de Blancco. Ya no recibe ni soporte ni más actualizaciones y, por ende aunque funciona bien, hay casos o discos duros que ya no están soportados y su uso se enfoca a nivel personal no en el ámbito de empresa. Más información en su web <https://dban.org/>.

1.2.5. Conclusiones

A día de hoy existen muchas alternativas de herramientas para obtener la información sobre un ordenador (Apartado 1.2.1), y también que realicen benchmarking (Apartado 1.2.2) o tests sobre los componentes (Apartado 1.2.3) o un borrado de datos (Apartado 1.2.4). Si se hace una pequeña búsqueda de las alternativas similares, se puede ver que la mayoría son herramientas muy concretas para un propósito específico, realmente hay pocas alternativas que engloben la obtención de la información del dispositivo para realizar un registro, conjuntamente con un diagnóstico del estado del mismo y un borrado de datos del dispositivo.

Aparte de las herramientas que son más completas y engloban tanto el registro como el diagnóstico conjuntamente con un sistema de gestión de dispositivos son en su gran mayoría software privativo. Por lo tanto si se descartan todas las opciones que no sean realmente software de código abierto, no hay una opción clara que englobe conjuntamente el registro y el diagnóstico del ordenador con un sistema de inventario.

En el ecosistema eReuse, se puede subir directamente toda esta información recopilada por la herramienta Workbench a un inventario web, Devicehub, sin necesidad de interacción por parte del usuario. El ecosistema desarrollado y mantenido por eReuse permite obtener información de los ordenadores de forma automática, usando la herramienta Workbench, y posteriormente mantener la trazabilidad de estos ordenadores durante lo largo de su vida útil hasta su reciclaje a través del

sistema de gestión de dispositivos Devicehub. Por desgracia, el software Workbench aún no está suficientemente maduro en cuanto a escalabilidad para estar en producción. Nos hemos encontrado ciertas situaciones en que la herramienta no es escalable y sería deseable tener una solución a estas situaciones. Como los recursos del proyecto eReuse son limitados, tenemos que priorizar los desarrollos según las necesidades reales y también la mejora que nos aportaría implementar dicha solución. Por eso también estamos buscando una manera de analizar y evaluar la mejora que nos aportaría implementar una solución propuesta o un desarrollo pendiente dentro del ecosistema eReuse.

2. Alcance del proyecto

Este proyecto tiene un alcance inicial bastante extenso aunque vamos a concretar un poco. De forma general nos vamos a enfocar en las claves para escalar un software de registro y diagnóstico de ordenadores. Pero más concretamente nos vamos a centrar en cómo podemos medir la escalabilidad de una herramienta de software como el Workbench. Para ello vamos a crear una metodología, y aplicar esta metodología sobre un conjunto de soluciones propuestas para unas situaciones concretas. También vamos a implementar las soluciones para validarlas técnicamente y medir su nivel de madurez tecnológica.

2.1. Objetivos

Este trabajo tiene como **objetivo inicial**, crear una metodología para valorar cuantitativamente la mejora de escalabilidad sobre una solución dada.

Como **segundo objetivo**, evaluar varias situaciones potencialmente interesantes, basadas en casos que se han encontrado dentro del proyecto eReuse.org y casos que muy posiblemente podrían ocurrir en un futuro cercano. Y definir una posible solución, que sea técnicamente y económicamente viable, para cada una de las situaciones evaluadas anteriormente.

Como **tercer objetivo**, aplicar la metodología creada anteriormente sobre el conjunto de soluciones propuestas para determinar una puntuación de mejora para cada solución. Este valor refleja el grado de escalabilidad que nos genera implementar la solución dentro de la herramienta Workbench.

Y como **objetivo final**, obtener una lista priorizada de soluciones y escoger una solución como prioritaria, la que nos puede suponer una mejora de escalabilidad mayor para las necesidades reales de la herramienta Workbench. Por último vamos a crear un prototipo funcional sobre la solución escogida, definiendo funcional como un prototipo que se pueda usar y cumpla con los requisitos definidos en la descripción de la solución escogida.

En resumen la finalidad de este trabajo es mostrar una forma de evaluar unas posibles soluciones a situaciones reales concretas de un software de registro de ordenadores, para poder tomar decisiones sobre que desarrollo priorizar, dependiendo de la mejora de escalabilidad que nos aporte cada solución. Este trabajo nos facilita la tarea de priorizar los desarrollos y poder definir con más rigurosidad un *roadmap*, es decir una hoja de ruta, que se adapte lo mejor posible a las necesidades reales del software. Este trabajo termina en el punto en que se escoge una solución como prioritaria, después de argumentar la elección usando una metodología, y crear un prototipo funcional para poder aplicar la solución en la vida real y así tener una aproximación más realista de la mejora que supone desarrollar la solución escogida. Este prototipo va a servir, en un futuro trabajo, como punto de partida para realizar la integración con el ecosistema eReuse.

2.2. Posibles obstáculos

Durante el proyecto pueden surgir ciertos imprevistos y/o dificultades durante la realización del mismo que impliquen modificaciones de metodología, de planificación temporal o de otras características del proyecto. Algunas de estas dificultades pueden ser:

- Adaptación al ecosistema eReuse y sobre todo a los sistemas vinculados a Workbench.
- Dificultad con las posibles tecnologías a trabajar durante el proyecto.
- Complejidad al buscar una solución flexible y escalable, ya que normalmente al buscar flexibilidad suele crecer la complejidad.
- Identificación de las necesidades reales del proyecto dentro de eReuse.
- Configuración de la infraestructura necesaria para el desarrollo del proyecto.
- Propuesta de solución que no se pueda llevar a cabo por falta de tiempo o recursos.
- Validar correctamente la viabilidad de una solución propuesta.
- Dificultad con la creación de un prototipo funcional.
- Posible reestructuración de la planificación por imprevistos no contemplados anteriormente.

2.3. Metodología

Para desarrollar este trabajo utilizaremos una metodología ágil. La metodología ágil es una práctica que promueve la iteración continua del desarrollo y las pruebas durante todo el ciclo de vida de desarrollo de un proyecto. Esta metodología nos permite ir modificando el proyecto con tal de adaptarnos a las posibles dificultades que surjan. A diferencia de una metodología tradicional, donde todas las fases están muy marcadas, las metodologías ágiles nos permiten adaptarnos al cambio con facilidad. Por lo tanto, las metodologías ágiles buscan gestionar de manera generalizada cómo se van completando las tareas.

Dentro de las metodologías ágiles existen una gran variedad de metodologías, pero nos vamos a centrar en la metodología *Scrum* y *Kanban*. *Scrum* es una metodología de desarrollo de software ágil iterativa e incremental para la gestión del desarrollo de productos. *Kanban* es una palabra japonesa que se compone de dos partes: Kan, que significa visual, y Ban, que hace referencia a tarjeta, de modo que como podemos deducir la metodología utiliza tarjetas para gestionar, de manera visual, la realización de determinados procesos y tareas.

En este proyecto vamos a usar una plataforma web que usa tanto *Scrum* como *Kanban* de código abierto y gratuita llamada *Taiga* [36]. *Taiga.io* es una plataforma de gestión de proyectos para *startups*, desarrolladores y diseñadores ágiles que quieren una herramienta simple, visualmente atractiva y fácil de usar. Dentro de la plataforma *Taiga.io* vamos a crear un conjunto de **épicas** que equivalen y engloban, tanto a los diferentes situaciones definidas por el equipo de eReuse, como las diferentes posibles soluciones a estas situaciones. Dentro de estas épicas vamos a crear las **historias de usuario** (*user stories*) donde se va a exponer toda la información necesaria para definir y evaluar bien las situaciones planteadas. Por último, también vamos a crear **tareas** que es la unidad más pequeña que podemos crear dentro de la plataforma *Taiga.io*, van a servir para definir pasos a seguir para poder conseguir los objetivos planteados dentro del proyecto.

Al final, al tener una correlación entre el conjunto de soluciones propuestas y las épicas creadas en el *Taiga.io*, podemos priorizar las épicas usando la metodología creada según el grado de escalabilidad que aporta según la situación que se nos plantee.

Para lograr los objetivos explicados en la sección 2.1, sería muy recomendable comenzar por una búsqueda analizando el estado del arte de la herramienta tal como aparece en la sección 1.2. Seguidamente definir bien el concepto de escalabilidad y como se aplica dentro del proyecto. Acotar bien las diferentes dimensiones sobre las cuales vamos a trabajar para evaluar la escalabilidad y también las situaciones que nos plantean desde eReuse.org sobre los cuales vamos a proponer posibles soluciones.

A continuación una breve explicación de las dimensiones que se van a tratar en esta metodología para calcular el potencial de escalabilidad:

- **Dimensión A: Volumen de dispositivos.**

La dimensión A se enfoca en la característica de observar desde el punto de vista del volumen de dispositivos a registrar y diagnosticar. Es decir, evaluar la capacidad de realizar muchos registros de ordenadores concurrentemente.

Para poder evaluar esta dimensión vamos a formular una pregunta: **¿Del 0 al 3, cuánto permite aumentar la solución propuesta la capacidad de registrar un gran volumen de dispositivos, es decir la capacidad de registrar muchos ordenadores a la vez?**

- **Dimensión B: Volumen de usuarios.**

La dimensión B se enfoca en la característica de observar desde el punto de vista del volumen de usuarios usando la herramienta Workbench. Es decir, evaluar la capacidad de asumir muchos usuarios nuevos usando la herramienta.

Para poder evaluar esta dimensión vamos a formularnos una pregunta: **¿Del 0 al 3, cuánto permite aumentar la solución propuesta la capacidad de en muy poco tiempo tener un gran número de usuarios nuevos usando la herramienta?**

- **Dimensión C: Diversidad de modelos (hardware diferente).**

La dimensión C engloba la característica de observar desde el punto de vista de la diversidad de modelos de ordenadores, modelos diferentes, a registrar y diagnosticar. Es decir, la capacidad de adaptarse y poder registrar correctamente diferentes modelos de ordenador con hardware variado.

Para poder evaluar esta dimensión vamos a formularnos una pregunta: **¿Del 0 al 3, cuánto permite aumentar la solución propuesta la capacidad de adaptar el software para registrar ordenadores diferentes (hardware distinto)?**

- **Dimensión D: Coste de registro y diagnóstico.**

La dimensión D se enfoca de forma general en los costes de registro de un ordenador, aunque vamos a dividirlo en tres subdimensiones más concretas, para no aumentar mucho la complejidad de la metodología vamos a englobarlas en una sola dimensión que se obtendría como media de las otras tres subdimensiones.

Para poder evaluar esta dimensión vamos a formularnos una pregunta: **¿Del 0 al 3, cuanto permite disminuir la solución propuesta el coste global de realizar un registro?**

- **Dimensión D1: Coste registro operario.**

La dimensión D1 como bien explica su nombre sería observar desde el punto de vista del coste que genera el operario o usuario que usa el software Workbench para realizar un registro de un ordenador. Para obtener un valor sobre esta subdimensión nos podemos plantear la siguiente pregunta: **¿Del 0 al 3, cuánto permite disminuir la solución propuesta el coste del operario al realizar un registro de un ordenador?**

- **Dimensión D2: Coste registro soporte técnico.** La dimensión D2 como bien explica su

nombre sería observar desde el punto de vista del coste que genera el soporte técnico para realizar correctamente un registro de un ordenador que anteriormente falló. Para obtener un valor sobre esta subdimensión nos podemos plantear la siguiente pregunta: **¿Del 0 al 3, cuánto permite disminuir la solución propuesta el coste del soporte técnico al realizar un registro que requiere su atención?**

- **Dimensión D3: Coste infraestructura/hardware.** La dimensión D3 sería poner el observador desde el punto de vista de recursos materiales o hardware necesarios para realizar el registro o dar uso al software Workbench. Dicho de otra forma, los costes de infraestructura que hay para usar la herramienta. Para obtener un valor sobre esta subdimensión nos podemos plantear la siguiente pregunta: **¿Del 0 al 3, cuánto te permite disminuir la solución propuesta el coste de infraestructura o material para usar la herramienta?**

Una vez definidas las dimensiones sobre las que vamos a trabajar, vamos a definir cómo calculamos el valor de mejora de escalabilidad. Vamos a usar varios grados divididos en **valor de mejora alto (3), medio (dos puntos), bajo(1) o nulo(0)**. Asignaremos uno de estos valores a cada una de las dimensiones explicadas anteriormente.

Al final esta metodología se crea para poder analizar de una forma mucho más rigurosa y clara el potencial de mejora de escalabilidad que nos puede aportar un desarrollo pendiente o una nueva propuesta de solución dada una situación concreta. Por último vamos a crear una tabla para cada solución propuesta. Así se podrá observar mejor las puntuaciones de cada solución, principalmente qué valor de mejora se obtiene sobre cada dimensión y la puntuación total de mejora de escalabilidad, tal como se muestra en la Figura 3.

Dimensiones	Valor de mejora
<i>(Dim A) Volumen Dispositivos</i>	Alto (3 puntos)
<i>(Dim B) Volumen Usuarios</i>	Medio (2 puntos)
<i>(Dim C) Diversidad Modelos</i>	Nulo (0 puntos)
<i>(Dim D) Coste global registro</i>	Bajo (1 puntos)
<i>(Dim D1) Coste registro operario</i>	Medio (2 puntos)
<i>(Dim D2) Coste registro mantenimiento</i>	Nulo (0 puntos)
<i>(Dim D3) Coste registro infraestructura</i>	Nulo (0 puntos)
Puntuación Total (sobre 12)	6 de 12 puntos

Figura 3: Tabla ejemplo - Mejora escalabilidad

3. Planificación temporal

Este proyecto está pensado inicialmente para tener **una duración de unas 18 semanas** aproximadamente, empezando desde principios de septiembre, más concretamente desde la semana del 7 de septiembre hasta finales de diciembre o principios de enero.

Durante el planteamiento inicial se calcula una dedicación en horas de alrededor de las **500 horas**, suponiendo que no habrá ningún obstáculo o imprevisto grave que haga rectificar algún aspecto importante del proyecto, teniendo así que dedicar más horas para finalizar el proyecto correctamente. Dentro de este cómputo de horas también está incluidas todas las tareas vinculadas a la materia de Gestión de Proyectos (GEP), que se realizaran durante el primer mes del proyecto. Estas tareas tiene como finalidad ayudar a definir correctamente el planteamiento y la estructura del proyecto y documentarlo desde el inicio, para tener una memoria del proyecto más pulida, estructurada y completa posible. Esto facilita posteriormente poder realizar el seguimiento, la finalización y defensa del proyecto con mucha más facilidad.

3.1. Descripción de tareas

3.1.1. Familiarización con el ecosistema eReuse

La primera tarea es **familiarizarse con el ecosistema eReuse**, empezando por probar y ver en qué estado están las herramientas que vamos a usar, o que van a intervenir de algún modo dentro del proyecto. También, qué tecnologías concretas se usan mayoritariamente y si puede ser un obstáculo trabajar con alguna de ellas. Sobre todo, el objetivo es entender cómo funciona internamente el software Workbench, para posteriormente poder dar la mejor solución para hacer este software escalable.

Más concretamente consiste en descargar y probar la herramienta Workbench, hacer todos los test pertinentes así como también ver su código y saber un poco como está estructurado en cuanto a código y funcionamiento.

El resultado es una descripción del ecosistema eReuse y las partes que lo componen, tal como se muestra en la sección 1. Esta tarea tiene una duración aproximada de unas **35 horas**.

3.1.2. Creación de una metodología para valorar la escalabilidad

Después de investigar y familiarizarse con el ecosistema eReuse, vamos a **crear una metodología para poder analizar y evaluar en escalabilidad una solución o épica concreta**. Estas soluciones o épicas responden a determinadas situaciones reales que se puede encontrar la herramienta Workbench y que se han ido planteando desde eReuse durante la fase de familiarización.

El resultado es la definición de todas las dimensiones sobre las que vamos a trabajar y se van a tener en cuenta para el cómputo de la puntuación, tal como se muestra en la sección 2.3. Al final, vamos a responder un conjunto de cuestiones para terminar obteniendo como resultado un valor de mejora que equivaldrá a un número entero para poder realizar la puntuación lo más objetiva posible. Esta tarea tendrá una duración aproximada de **65 horas**.

3.1.3. Evaluación de las situaciones planteadas por el equipo de eReuse

Esta tarea consiste en **evaluar las distintas situaciones** reales en las que se va a encontrar la herramienta Workbench y sobre las que sería deseable tener una solución escalable para solventar dichas situaciones.

Dichas situaciones vienen dada por la información proporcionada por el equipo de eReuse.org de situaciones reales extraídas de pruebas anteriores con otros agentes (empresas/entidades/organizaciones).

El resultado es una lista de situaciones con una definición acotada de en que consiste cada una de ellas, tal como se muestra en la sección 5, como pueden ser las condiciones que se van a dar en cada caso y que se tienen que cumplir para solventar cada situación. Esta tarea tiene una duración estimada en **45 horas**.

3.1.4. Definición de una posible solución sobre cada situación evaluada

Esta tarea consiste en **definir una propuesta de solución para cada una de las situaciones** descritas en la sección 5. Primero debemos tener claro los puntos a solventar en cada situación y después investigar, analizar y valorar que tecnologías, herramientas o cambios serian los más adecuados para aplicar sobre la herramienta en cada una de las situaciones.

El resultado es un conjunto de posibles soluciones con una definición acotada de en qué consiste cada una de estas soluciones y también en que aspectos se podría realizar una mejora más cuantitativa en cuanto a escalabilidad, tal como se muestra en la sección 6. Esta tarea será probablemente la tarea más larga dentro de este trabajo y por eso tiene una duración estimada en **110 horas**.

3.1.5. Comprobación de la viabilidad de las soluciones propuestas

En esta tarea se comprueba la **viabilidad de las soluciones propuestas** durante la tarea 3.1.4. Se hacen un conjunto de pruebas para verificar y asegurar que realmente es una solución que se pueda realizar a la práctica. Para asegurar esto, vamos a realizar una prueba de concepto para cada una de las soluciones y así garantizar su viabilidad. En caso de imprevistos también vamos a explicar cómo se han afrontado estas dificultades.

El resultado será una explicación de las pruebas de concepto realizadas para comprobar que la solución propuesta es viable. Esta tarea también es muy importante y por eso se dedica una parte importante del proyecto, más concretamente una estimación de **50 horas**.

3.1.6. Aplicación de la metodología sobre las soluciones válidas

Esta tarea consiste en **aplicar la metodología** creada sobre cada una de las soluciones planteadas anteriormente para calcular su potencial de mejora de escalabilidad.

Una vez se tenga una visión suficientemente amplia de las posibles soluciones planteadas, se aplicará la metodología sobre las soluciones para obtener una tabla comparativa de puntuaciones.

El resultado consistirá en a partir de la tabla, realizar una lista ordenada por mayor puntuación de mejora de escalabilidad, y escoger cual es la solución prioritaria para desarrollar primero. Esta tarea en cuestión puede tener una duración variable, dependiendo del número de soluciones encontradas, tiene una duración estimada de **30 horas**.

3.1.7. Creación de un prototipo de la solución escogida como prioritaria

En esta tarea se va a **crear un prototipo funcional de la solución escogida como prioritaria**. Definiendo funcional como un prototipo que se pueda usar y cumpla con los requisitos definidos en la descripción de la solución escogida.

Aunque no está previsto realizar su integración dentro del ecosistema eReuse durante este trabajo, si haremos el prototipo que nos dará un punto de partida para en un futuro trabajo llevar a cabo la integración dentro del ecosistema eReuse.

El resultado será un prototipo que se pueda ejecutar y comparar para ver la mejora real que nos aporta implementar la solución escogida. esta tarea es muy importante y por eso se dedica una parte importante del proyecto, más concretamente una estimación de aproximadamente **75 horas**.

3.1.8. Documentación y preparación de la defensa

La última tarea del proyecto será terminar de **redactar la documentación** una vez finalizada la parte de desarrollo, de las pruebas y también de posibles mejoras sobre el funcionamiento de la aplicación implementada.

El resultado será un documento que servirá de memoria del proyecto y que será expuesto para la defensa del proyecto delante de un tribunal de profesores de la facultad. Esta tarea se estima una dedicación de **30 horas**.

3.2. Estimaciones

A continuación la Figura 4 donde se puede ver la tabla de estimaciones con el conjunto de tareas a realizar durante el trabajo y posteriormente el diagrama de Gantt representado en la Figura 5.

Código	Descripción de la tarea	Tiempo estimado	Fecha Inicio	Fecha Fin
T0	Gestión del proyecto	50	07/09/20	20/11/20
T1	Familiarización con el ecosistema eReuse	35	07/09/20	30/09/20
T2	Creación de una metodología para valorar la escalabilidad	65	21/09/20	23/10/20
T3	Evaluación de las situaciones planteadas por el equipo de eReuse	45	28/09/20	09/10/20
T4	Definición de una posible solución sobre cada situación evaluada	110	09/10/20	24/11/20
T5	Comprobación de la viabilidad de las soluciones propuestas	50	25/11/20	07/12/20
T6	Aplicación de la metodología sobre las soluciones validas	30	09/12/20	15/12/20
T7	Creación de un prototipo de la solución escogida como prioritaria	75	16/12/20	04/01/21
T8	Documentación y preparación de la defensa	30	02/11/20	04/01/21
	Total	490		

Figura 4: *Tabla de estimaciones de las tareas*

3.3. Diagrama de Gantt

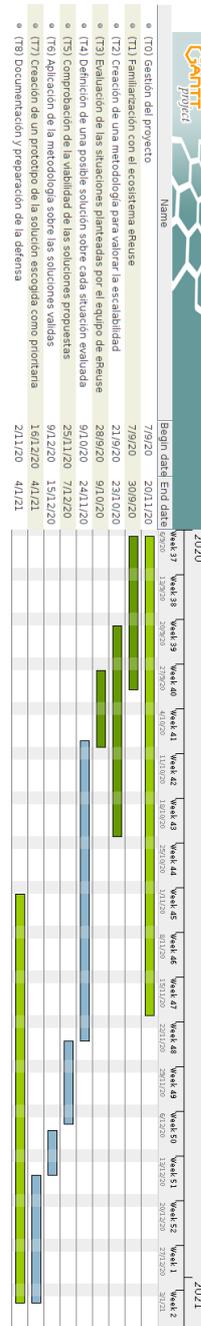


Figura 5: Diagrama de Gantt

3.4. Recursos usados

Dentro de los recursos necesarios para llevar a cabo este proyecto, vamos a diferenciar entre recursos técnicos, principalmente software o infraestructura necesaria, y recursos humanos, es decir todas aquellas personas implicadas dentro del proyecto y que sin ellas no sería posible realizarlo.

3.4.1. Recursos técnicos

- **GitHub** [37]: Plataforma donde vamos a consultar el código de las herramientas de eReuse y en caso necesario también vamos a subir código hecho para el trabajo. Así podemos mantener un control de versiones y también compartirlo fácilmente con el equipo de eReuse.
- **Taiga** [36]: Es una plataforma de gestión de proyectos para *startups* y desarrolladores y diseñadores ágiles que precisan de una herramienta simple y fácil de usar.
- **Workbench** [5]: Software de eReuse.org escrito en Python 3 que captura las características de hardware (como el número de serie, modelo, fabricante, etc.) de los ordenadores y sus componentes, también realiza pruebas para obtener un pequeño diagnóstico del estado del ordenador, y borra los datos de forma segura en caso que sea necesario.
- **Workbench Live** [17]: Software de eReuse.org que genera una imagen live ISO basada en un sistema base Debian 9 que autoejecuta la herramienta Workbench al arrancar la imagen.
- **Devicehub** [15]: Es una aplicación web RESTful desarrollada en Python 3 por eReuse.org para realizar gestión de dispositivos, comúnmente conocido como un inventario.
- **PyCharm** [38]: Es un IDE (Entorno de desarrollo integrado) enfocado al lenguaje Python con el cual se ha realizado toda la parte de lectura y desarrollo del código dentro del proyecto.

3.4.2. Recursos humanos

- **Desarrollador principal**: El autor de este proyecto será la única persona que desarrolle el proyecto, así como el encargado de cumplir con todas los hitos definidos, documentar el proyecto y finalmente también presentarlo.
- **Director del proyecto**: El director de este proyecto es David Franquesa Griso, su papel principal es supervisar el cumplimiento de los objetivos definidos y el calendario establecido para el desarrollo de este proyecto. También en caso de necesidad, dar los recursos necesarios para poder proceder con el desarrollo del proyecto.
- **Supervisor del TFG**: En este caso es Leandro Navarro Moldes. Encargado de supervisar a nivel técnico y académico el cumplimiento de los requisitos necesarios dentro del proyecto, así como resolver posibles dudas durante el transcurso del proyecto y también corregir el enfoque del proyecto de forma dinámica en caso necesario.
- **Equipo de eReuse**: Son el equipo encargado de suministrar las tecnologías requeridas, gestionar y resolver posibles dudas técnicas dentro del funcionamiento de las herramientas de eReuse. También se encargan de realizar el aprovisionamiento de la infraestructura requerida para el desarrollo del proyecto.

4. Gestión económica

4.1. Presupuesto

Dentro del presupuesto vamos a identificar los costes por tipo de recursos necesarios para llevar a cabo este proyecto. Vamos a diferenciar entre recursos de software, recursos de hardware o infraestructura necesaria, y recursos humanos. Es decir, todas aquellas personas implicadas dentro del proyecto y que sin ellas no sería posible realizarlo.

4.1.1. Recursos Software

Vamos a exponer este tipo de recurso, remarcando que aunque si se usan bastantes recursos de software no suponen ningún coste dentro del presupuesto. Al tratarse de código abierto podemos realizar un uso sin un coste añadido de licencias, excepto el IDE PyCharm que tenemos una licencia educativa gratuita.

- **GitHub** [37].
- **Taiga** [36].
- **Workbench** [5].
- **Workbench Live** [17].
- **Devicehub** [15].
- **PyCharm** [38].

4.1.2. Recursos Hardware

Dentro de los requisitos de equipos e infraestructura del proyecto hay principalmente, un ordenador portátil que se usará tanto para realizar el desarrollo necesario como las pruebas de viabilidad de las soluciones propuestas.

Después también vamos a usar temporalmente otros equipos como clientes, sobre todo por la parte de pruebas, pero no vamos a ponerlo dentro del presupuesto porque van a ser equipos reutilizados, que solo vamos a necesitar temporalmente y que no son necesarios para el producto en sí.

Por último, añadir que los recursos hardware podrían aumentar durante la definición de las soluciones propuestas, aunque las soluciones propuestas tienen que ser económicamente viables y no requerir muchos recursos, aun así vamos a necesitar materiales varios, como un monitor, una memoria USB con al menos un GB de capacidad, cables varios (como cables de red, vídeo, alimentación, etc.) y también herramientas varias que no vamos a especificar porque realmente no son relevante para el trabajo.

4.1.3. Recursos humanos

Necesitamos principalmente los siguientes roles o personal para poder llevar a cabo el proyecto.

- **Desarrollador principal:** Encargado de la correcta implementación del proyecto, será también la única persona que desarrolle el proyecto, así como el encargado de cumplir con todas las actividades definidas dentro del Gantt. Va a suponer un coste de aproximadamente de **10 €/hora**. La dedicación de las tareas planificadas dentro del proyecto es de aproximadamente 490 horas, suponiendo un coste total de **4900 euros**.
- **Equipo de eReuse:** Son el equipo humano encargado de suministrar las tecnologías requeridas, gestionar y resolver posibles dudas técnicas dentro del funcionamiento de las herramientas de eReuse. También se encargan de realizar el aprovisionamiento de la infraestructura requerida para el desarrollo del proyecto. Con un coste de **20 €/hora**. Su dedicación de horas en total será bastante flexible dependiendo de cómo vaya transcurriendo el proyecto y las necesidades que surjan, pero vamos a computar unas 20 horas, suponiendo un coste de **400 euros**.
- **Director del proyecto:** Encargado de supervisar el cumplimiento de los objetivos definidos y el calendario establecido para el desarrollo de este proyecto. También se encarga de corregir el enfoque del proyecto de forma dinámica en caso de ser necesario. Va a realizar una dedicación de 20 horas, con un coste de **30 €/hora**, en total **600 euros**.

4.1.4. Contingencias

Durante el proyecto pueden surgir ciertos imprevistos y/o dificultades que impliquen modificaciones de metodología, de la planificación temporal e incluso de otras características del proyecto que nos pueden suponer un sobrecoste importante si no se ha tenido en cuenta desde el inicio del proyecto.

Principalmente vamos a prevenir dos imprevistos reflejados dentro del presupuesto. Uno sería durante la definición de las posibles soluciones sobre la escalabilidad del Workbench se opte por una solución que más tarde sea muy difícil de validar técnica o económicamente. Y el segundo imprevisto es que durante la fase de implementación del prototipo, aunque se haya implementado correctamente la solución seleccionada, se tengan que realizar cambios severos sobre la implementación para obtener un prototipo funcionalmente válido.

4.1.5. Presupuesto final

A continuación se muestra una tabla resumen con los costes totales desglosados tanto de las estimaciones de los recursos humanos, descritos en 4.1.1, necesarios para realizar las diferentes tareas, en verde definido como total CPA (Total de costes de personal por actividad), como también los recursos materiales, descritos en 4.1.2, en azul denominados CG (Costes imputados genéricamente). También aparece la contingencia aprobada, en amarillo y el coste relativo de los imprevistos descritos anteriormente en el apartado 4.1.4 en rojo.

Actividad	Importe (€)
(T0 / T8) Gestión del proyecto / Documentación	800
(T1) Familiarización con el ecosistema eReuse	350
(T2) Creación de una metodología para valorar la escalabilidad	650
(T3) Evaluación de las situaciones planteadas por el equipo de eReuse	450
(T4) Definición de una posible solución sobre cada situación evaluada	1100
(T5) Comprobación de la viabilidad de las soluciones propuestas	500
(T6) Aplicación de la metodología sobre las soluciones validas	300
(T7) Creación de un prototipo de la solución escogida como prioritaria	750
Dirección del proyecto	400
Consultoría con el equipo eReuse	600
Total CPA	5900
Amortización de equipo e infraestructura (Hardware)	150
Amortización de Software	0
Costes de material	50
Total CG	200
Total Costes (Total CPA + Total CG)	6100
Contingencia	915
Total CD+CI +Contingencia	7015
I1 (Coste=500; riesgo=15%)	75
Total Imprevistos:	75
TOTAL:	7090

Figura 6: Tabla con el presupuesto final del proyecto

4.1.6. Control de gestión

Dentro del control de la gestión destacar que la herramienta *Taiga*, ya mencionada dentro del apartado 4.1.1, nos permite llevar una organización ágil y eficiente sobre el desarrollo del proyecto, y también ver rápidamente si estamos cumpliendo los objetivos de la forma esperada. Aun así, vamos a añadir una **contingencia del 15%** de los costes sin los imprevistos, para asegurar que en caso de tener algún obstáculo mayor no previsto, tengamos un pequeño margen de maniobra y podamos realizar alguna hora extra para resolverlo.

5. Evaluación de las situaciones planteadas por el equipo de eReuse

Vamos a evaluar las distintas situaciones reales en las que se va a encontrar la herramienta Workbench y sobre las que sería deseable tener una solución escalable. Dichas situaciones vienen dadas por la información proporcionada por el equipo de eReuse.org de situaciones reales extraídas de pruebas y proyectos anteriores con otros agentes (empresas/entidades/organizaciones).

Para entrar en más detalle vamos a definir las diferentes situaciones que se nos han planteado después de analizar e investigar el estado actual de la herramienta Workbench y ver las necesidades reales del producto para que sea realmente escalable, también añadiendo en todo momento las necesidades que nos marcan desde la vista de producto y sobre todo valorando las mejoras como se va a enfocar dentro de su uso en producción. También se va a exponer cómo se tratan actualmente estas situaciones y qué sería deseable solventar en cada situación.

A continuación se explica cada una de las situaciones evaluadas, cómo funciona actualmente y qué limitaciones aparecen:

- **Situación A: Registrar varios ordenadores de forma simultánea.**

Tenemos un **gran volumen de ordenadores similares pendientes de procesar**. En este contexto vamos a buscar de obtener una solución que permita, de una forma relativamente sencilla y rápida, instalar y ejecutar el Workbench en varios ordenadores paralelamente.

Actualmente la herramienta realiza los registros individualmente, y aunque se puede realizar varios registros a la vez, hay la limitación que hacen falta varias memorias USB y grabar la ISO dentro de cada uno de ellos. Igualmente solo se pueden registrar tantos ordenadores a la vez como memorias USB haya con la imagen ISO del Workbench dentro.

En este caso el indicador de impacto (KPI) será el número de ordenadores que se pueden registrar por operario (usuario) en un espacio de tiempo finito, como por ejemplo una hora. Una posible solución se describe más adelante en la sección [6.1](#).

- **Situación B: Asegurar el registro de cada ordenador individualmente.**

Tenemos un **volumen pequeño de ordenadores pero muy diferentes entre si, con modelos y fabricantes muy diversos**. Buscamos una solución o mecanismo que permita en caso de fallo durante el registro al inventario Workbench, poder obtener correctamente los datos de prioridad alta, es decir los datos que nos permiten identificar de forma única un ordenador.

De momento este proceso no es precisamente ágil. Actualmente el Workbench se puede ejecutar con un parámetro extra llamado 'debug' que guarda todos las salidas que usa el propio Workbench para generar el snapshot (json) del ordenador, que termina registrado dentro del inventario Devicehub. Partiendo de la base que podemos obtener los outputs del lshw y el hwinfo para recrear el snapshot que ha fallado anteriormente, vamos a diseñar nuestro sistema de depuración ágil automatizando todos aquellos pasos que nos supongan una mejora de escalabilidad y que sea viable implementar.

Un ordenador compatible con el Workbench, es aquel que se obtiene correctamente los datos de prioridad alta, que nos permite identificar de forma única el ordenador.

Una posible solución se describe más adelante en la sección 6.2.

- **Situación C: Proveer una imagen ISO específica para cada usuario.**

Tenemos un **volumen considerable de diferentes usuarios, nuevos dentro del sistema, que quieren ejecutar la herramienta Workbench** y subir el registro del ordenador al inventario Devicehub. Actualmente este proceso no es escalable porque se genera una imagen ISO específica para cada usuario con sus credenciales.

Aquí vamos a buscar una solución que nos permita ejecutar la herramienta sin tener que crear una imagen nueva de la herramienta para cada usuario. Es decir proveer una misma imagen ISO para todos los usuarios y que, usando esta ISO genérica, se pueda modificar para asociarla con el usuario en lugar de generar una imagen ISO nueva particular para cada usuario. Esto supone un tiempo de trabajo y computación que no permite la herramienta ser escalable en cuanto a crecimiento de nuevos usuarios

En este caso, el indicador de impacto (KPI) sería la diferencia de tiempo de ejecución entre el método que se usa actualmente y el método propuesto por la solución para crear una ISO específica asociada a un usuario y que los *snapshots* generados con ella se pueden asociar a ese usuario. Una posible solución se describe más adelante en la sección 6.3.

- **Situación D: Mostrar información relevante sobre la causa del error al usuario.**

Tenemos un **ordenador concreto que falla durante la ejecución del Workbench, antes de subir la información del registro al inventario Devicehub**. Actualmente en caso de fallo durante la ejecución siempre imprime por pantalla el mismo error, casi sin dar información sobre la causa del problema al usuario. Esto nos impide ser escalables en cuanto a mantenimiento y atención técnica para poder detectar y solucionar errores de una forma ágil. Al no reportar los errores con un mínimo de detalle, se hace mucho más difícil recrear y depurar el fallo.

En este caso se plantea una solución que por un lado acote en la cual de las cuatro funciones clave del Workbench, explicadas en la sección 1.2, ha fallado. También ha de mostrar al usuario información realmente relevante para poder reportar el error de forma más detallada y así ganar agilidad tanto para recrear el fallo como para encontrar una posible solución.

En este caso, el indicador de impacto (KPI) es cada usuario sepa la causa del fallo y pueda encontrar una solución (reportar, permitir corregir).

Una posible solución se describe más adelante en la sección 6.4.

- **Situación E: Registrar un ordenador con un hardware virtualizado .**

Tenemos **muy pocos ordenadores para pruebas con hardware diferente** y queremos intentar registrar ordenadores muy diversos, incluso con arquitecturas diferentes, sin tener que obtener y montar físicamente todos estos modelos concretos de ordenadores.

En esta situación concreta vamos a buscar una solución que nos permita registrar un ordenador con un hardware virtualizado y asegurar que efectivamente ese modelo de ordenador se puede añadir como compatible dentro de la herramienta Workbench.

En este caso, el indicador de impacto (KPI) sería que se puedan preparar las herramientas para

que pueda funcionar a la primera en un nuevo modelo antes de tenerlo disponible.
Una posible solución se describe más adelante en la sección 6.5.

Antes del inicio del proyecto solo se tuvo en cuenta la situación *A*. Después durante el trabajo juntamente con el equipo de eReuse se revisó el feedback o la información que se tenía sobre situaciones reales de usuarios usando la herramienta y nos dimos cuenta de que la mejora de escalabilidad de la herramienta Workbench por los usuarios actuales no se correspondía con el contexto de un gran volumen de ordenadores a procesar sino con otros contextos con un potencial de repetición mucho más alta por los usuarios actuales y en los que se podría aportar una mejora de escalabilidad mayor.

También debemos poner de manifiesto que hay condicionamientos entre las diferentes situaciones. Por ejemplo si no podemos registrar modelos muy diferentes o no podemos mantener la creación de nuevos usuarios que usen la herramienta, la situación *A* no es óptima sin tener aún las otras situaciones *B* y *C* cubiertas.

Una vez expuestas las diferentes situaciones, vamos a ver que posibles soluciones encontramos y cómo podemos proceder con cada una de ellas. Dentro de todas las posibles soluciones expuestas vamos a comprobar la viabilidad de todas ellas y escoger la solución con mayor potencial de mejora usando la metodología creada.

6. Posibles soluciones sobre la escalabilidad del Workbench

Vamos a crear una propuesta de solución para cada uno de las situaciones descritas en la sección 5. Estas soluciones están enfocadas a mejorar la escalabilidad de la herramienta. Es decir se trata de demostrar que se pueden llevar a la práctica todas las soluciones planteadas en la medida de lo posible por tiempo y recursos.

Antes de empezar con la explicación de las soluciones, puntualizar que una solución ofrece más o menos escalabilidad en función del contexto o situación en donde se aplique. Si el perfil de reacondicionadora de eReuse, usuario actual del sistema, fueran de las que procesa miles de dispositivos al mes de solo cinco o seis modelos diferentes entonces las soluciones a implementar serian otras. La realidad es que el usuario del Workbench es una reacondicionadora pequeña que procesa poco volumen de dispositivos y que además este es muy diverso. Además, la naturaleza de las reacondicionadoras es tener operarios que son personas en programas inserción laboral por lo que se forman unos meses y después se van a otro sitio. Esto genera unas limitaciones claras, principalmente que la gran mayoría de usuarios del Workbench no van a ser muy experto, por lo que se busca soluciones que faciliten al usuario poder usar la herramienta de una forma rápida y sin conocimientos técnicos previos. También a la vez es muy importante que no aumente el coste de infraestructura, mantenimiento o soporte técnico.

Después de la descripción de cada solución, se va a explicar si cada solución es viables técnicamente y económicamente, es decir vamos a hacer un conjunto de pruebas para verificar y asegurar que realmente es una solución que se pueda realizar en la práctica.

A continuación se explica cada una de las soluciones, en qué consisten, qué pruebas se han hecho para verificar la viabilidad y el impacto que tienen con respecto a la escalabilidad:

6.1. Solución A: Distribuir imagen ISO del Workbench por red usando un servidor PXE

6.1.1. Descripción de la solución

Partiendo del contexto de la situación A explicada anteriormente, vamos a proponer una solución basada en la distribución de una imagen ISO por red. Más concretamente usar un servidor de PXE para distribuir la imagen ISO que autoejecuta el Workbench por red simultáneamente en varios ordenadores.

6.1.2. Viabilidad de la solución

En esta solución en concreto ya tenemos un precedente dentro del ecosistema eReuse. Se hizo una instalación a medida y desatendida del sistema Debian usando un conjunto de scripts en Bash para instalar las dependencias y todos los paquetes necesarios para tener un servidor PXE funcional para el Workbench. Actualmente esta solución está obsoleta, ya que no es flexible ni escalable, porque está enfocada para funcionar sobre un hardware concreto y con unos costes altos tanto de instalación como de mantenimiento. Por eso, aunque es viable técnicamente realizar esta solución por el precedente que ya hay dentro del proyecto eReuse, será difícil que sea la solución escogida como prioritaria porque no es económicamente viable.

6.1.3. Aplicación de la metodología

Respecto al cálculo del potencial de mejora de escalabilidad de la solución, vamos a usar la metodología creada anteriormente en la sección 2.3 representándola en una tabla. Primero tenemos que matizar que se trata de una solución enfocada para la situación *A*, aunque al compararlo con todas las otras situaciones podremos ver si coincide el potencial de mejora con alguna de estas, aunque la solución inicialmente no está pensada para resolver otras situaciones.

Vamos a poner un valor sobre cada dimensión, empezando por la **dimensión A** que hace referencia al volumen de dispositivos que en este caso como la solución *A* nos va a permitir registrar varios ordenadores paralelamente va a tener un valor de mejora **Alto** que equivale a tres puntos.

Para la **dimensión B** que sería el volumen de usuarios, en este caso el valor de mejora será **Nulo**, es decir cero puntos, ya que para cada nuevo usuario tendríamos que realizar la instalación del servidor PXE y esto impide que haya ninguna mejora de escalabilidad.

En la **dimensión C**, si nos preguntamos cuánto permite aumentar la solución propuesta la capacidad de registrar hardware distinto, la respuesta sería un valor **Nulo**, es decir cero puntos, porque la solución no aumenta en ningún caso la capacidad de registrar modelos muy distinto, sino de poder registrar muchos dispositivos a la vez.

Por último, la **dimensión D** que está compuesta por tres subdimensiones, la **dimensión D1** referente al coste del registro por operario tendría un valor **Medio** (dos puntos), ya que al ser una solución por red, solo tendría que arrancar los ordenadores por red para usar la herramienta Workbench. En cambio, la **dimensión D2** que es el coste del registro por soporte técnico, si nos preguntamos cuánto disminuirá el coste de mantenimiento con la solución *A*, la respuesta sería **Nulo**, es decir cero puntos, ya que la solución propuesta no reduce este coste en absoluto, de hecho seguramente lo aumenta. Y la **dimensión D3** referente al coste de infraestructura que en este caso va a aumentar al requerir más recursos como de entrada un ordenador que nos sirva como servidor, por lo tanto valor **Nulo** (cero puntos). Si hacemos la media redondeada de las subdimensiones D1, D2 y D3 nos sale un total de dos puntos divididos entre tres subdimensiones redondeado tendría como media un punto que equivale a un valor **Bajo** de mejora.

Una vez argumentado los valores asignados en cada dimensión para la solución *A*, vamos a mostrar una tabla general con el resultado del valor de mejora sobre cada dimensión para la situación *A*:

Potencial de mejora de la solución A en la situación A	
Dimensiones	Valor de mejora
<i>(Dim A) Volumen Dispositivos</i>	Alto (3 puntos)
<i>(Dim B) Volumen Usuarios</i>	Nulo (0 puntos)
<i>(Dim C) Diversidad Modelos</i>	Nulo (0 puntos)
<i>(Dim D) Coste registro global</i>	Bajo (1 punto)
<i>(Dim D1) Coste registro operario</i>	Medio (2 puntos)
<i>(Dim D2) Coste registro mantenimiento</i>	Nulo (0 puntos)
<i>(Dim D3) Coste registro infraestructura</i>	Nulo (0 puntos)
Puntuación Total (sobre 12)	4 de 12 puntos

Figura 7: Tabla Metodología (Situación A - Solución A)

6.1.4. Resultados

Como resultados de esta solución tenemos una puntuación de mejora de escalabilidad de **cuatro puntos sobre 12 para la solución A**. Hemos probado que es una solución viable técnicamente porque tenemos un precedente que lo demuestra con una versión antigua. A falta de ver las puntuaciones de las otras soluciones y poderlas comparar con estos resultados, vamos a decir que no es realmente una solución que aporte una mejora de escalabilidad significativa, ya que se encuentra por debajo de los seis puntos sobre 12 que es la mitad de los puntos que se pueden obtener en total.

6.2. Solución B: Sistema de depuración para la mejora continua de la herramienta

6.2.1. Descripción de la solución

Partiendo del contexto de la situación *B* explicada anteriormente, vamos a proponer una solución basada en agilizar el proceso de depuración de un nuevo modelo de ordenador para que pase a ser compatible para el Workbench. Este sistema de depuración va a permitir de una forma eficiente encontrar fallos en el registro de nuevos y diferentes modelos de ordenadores.

Más concretamente la solución se basa en tener un flujo de trabajo bien definido para agilizar al máximo todo el proceso de depuración para modificar el código y poder identificar correctamente un modelo nuevo de ordenador. Esta solución se basa en que podemos recuperar el snapshot enviado al inventario Devicehub ya que actualmente se guardan los json que fallan en un servidor. A partir de este json con toda la información del campo 'debug', podemos recrear el registro del Workbench, pero en un entorno controlado y sin necesidad de tener acceso físico al modelo de ordenador concreto.

6.2.2. Viabilidad de la solución

Para comprobar la viabilidad técnica de esta solución hemos hecho una prueba de concepto cogiendo un modelo de ordenador que no fuera compatible aún con el Workbench. Más concretamente nos han proporcionado desde eReuse, un snapshot erróneo para poder recrear todo el proceso de depuración definido para ver que en ningún punto hay un obstáculo técnico que nos impida llevar a la práctica la solución propuesta y también para poder responder con más criterio las preguntas planteadas en la metodología creada que vamos a aplicar en el próximo apartado. Se ha llegado a usar la solución parcialmente con ordenadores reales ya que la primera parte del campo 'debug' y la recuperación de la información de registro, es decir los *snapshots*, eran características que ya estaban implementadas previamente.

6.2.3. Aplicación de la metodología

Respecto al cálculo del potencial de mejora de escalabilidad de la solución, vamos a usar la metodología creada anteriormente en la sección 2.3 representándola en una tabla. Primero tenemos que tener en cuenta de que es una solución enfocada para la situación *B*. Es decir, la solución inicialmente no está pensada para resolver otras situaciones, aunque al compararlo con las otras situaciones podremos ver si coincide el potencial de mejora con alguna de estas.

Vamos a poner un valor sobre cada dimensión, empezando por la **dimensión A** que hace referencia al volumen de dispositivos que en este caso como la solución *B* nos va a permitir registrar nuevos modelos de ordenador de una forma escalable, es decir el impacto sobre la capacidad de registrar un gran volumen de dispositivos es **Bajo**, que equivale a uno punto, ya que la solución se centra en un corregir un modelo concreto de ordenador. Por tanto, tendrá un pequeño impacto en los casos en que no permite registrar ese modelo concreto, en los otros caso no va a permitir registrar muchos ordenadores a la vez.

Para la **dimensión B** que sería el volumen de usuarios. En este caso, al plantearnos la pregunta pertinente la respuesta sería un valor **Bajo** que equivale a un punto, ya que no impacta de forma significativa sobre el volumen de usuarios, pero si permite tener un volumen mayor de usuarios porque la solución nos va a permitir depurar de una forma ágil cuando ocurre un error al registrar el ordenador en el inventario Devicehub.

En la **dimensión C**, si nos preguntamos cuánto permite aumentar la solución propuesta la capacidad de registrar hardware distinto, la respuesta sería un valor **Alto**, es decir tres puntos, ya que la solución permite precisamente aumentar el número de modelos diferentes compatibles con la herramienta Workbench de forma escalable.

Por último, la **dimensión D** que está compuesta por tres subdimensiones, la **dimensión D1** referente al coste del registro por operario tendría un valor **Medio** (dos puntos), ya que la solución nos va a permitir aumentar el número de modelos diferentes compatibles con la herramienta. Esto va a disminuir bastante la probabilidad de que el operario reciba un error al registrar ordenadores nuevos o diferentes, y esto tiene un impacto considerable sobre el coste del registro por operario. Similar a lo expuesto antes tenemos la **dimensión D2** que es el coste del registro por soporte técnico. En este caso el valor de mejora será **Alto** (tres puntos) porque la solución está enfocada a mejorar la depuración en caso de error al registrar un ordenador y esto impacta fuertemente en el coste del soporte técnico que es el encargado de realizar dicha depuración. Y la **dimensión D3** referente al coste de infraestructura o reducción de los materiales requeridos para usar la herramienta. Podemos responder que la solución tendrá un valor de mejora **Nulo**, es decir cero puntos, ya que es este caso vamos a tener los mismos coste para usar la herramienta. Si hacemos la media redondeada de las subdimensiones D1, D2 y D3 nos sale un total de cinco puntos divididos entre tres subdimensiones redondeado tendría como media dos puntos que equivale a un valor **Medio** de mejora.

Una vez argumentado los valores asignados en cada dimensión para la solución *B*, vamos a mostrar una tabla general con el resultado del valor de mejora sobre cada dimensión para la situación *B*:

Potencial de mejora de la solución B en la situación B	
Dimensiones	Valor de mejora
<i>(Dim A) Volumen Dispositivos</i>	Bajo (1 punto)
<i>(Dim B) Volumen Usuarios</i>	Bajo (1 punto)
<i>(Dim C) Diversidad Modelos</i>	Alto (3 puntos)
<i>(Dim D) Coste registro global</i>	Medio (2 puntos)
<i>(Dim D1) Coste registro operario</i>	Medio (2 puntos)
<i>(Dim D2) Coste registro mantenimiento</i>	Alto (3 puntos)
<i>(Dim D3) Coste registro infraestructura</i>	Nulo (0 puntos)
Puntuación Total (sobre 12)	7 de 12 puntos

Figura 8: Tabla Metodología (Situación B - Solución B)

6.2.4. Resultados

Como resultados de esta solución tenemos una puntuación de mejora de escalabilidad de **siete puntos sobre 12 para la solución B**. Hemos probado que es una solución viable técnicamente y económicamente, ya que hemos realizado una prueba de concepto con un snapshot erróneo. A falta de ver las puntuaciones de las otras soluciones y poderlas comparar con estos resultados, de momento podemos decir que podría ser uno de las soluciones candidatas a ser escogida como prioritaria, ya que supera la puntuación de la solución anterior y aparte es una solución que aporta una mejora de escalabilidad significativa, ya que se encuentra por encima de los seis puntos sobre 12 que es la mitad de los puntos que se pueden obtener en total.

6.3. Solución C: Modificar la imagen ISO genérica para cada usuario específico

6.3.1. Descripción de la solución

Partiendo del contexto de la situación C explicada anteriormente, vamos a proponer una solución basada en modificar una imagen ISO genérica del Workbench para convertirla en una específica de un usuario sin crear una nueva imagen ISO desde cero.

Más concretamente la solución consiste en montar la imagen ISO genérica y modificar las líneas donde se definen las credenciales del usuario. Una vez modificadas dichas líneas se ha de regenerar la ISO pero guardando los cambios realizados. Así se obtiene una ISO específica para un usuario partiendo de una ISO genérica ya creada y ahorrando mucho tiempo de computación generando imágenes ISO. De esta forma vamos a mejorar drásticamente la escalabilidad en cuanto a altas de nuevos usuarios de la herramienta.

Existen varias opciones igualmente válidas, pero en este caso hemos usado una herramienta llamada *xorriso* [39] que es un paquete de software dentro del proyecto GNU mantenido por la Free Software Foundation (FSF) [40]. Con esta herramienta de código libre podemos abrir y extraer el contenido de la imagen ISO. Dentro de este contenido nos interesa modificar los datos locales de la ISO que en este caso se encuentran en formato *squashfs* que es un sistema de archivos de únicamente lectura comprimido para Linux. Para poder acceder a estos datos que en este caso se encuentran dentro de la carpeta 'live', concretamente el fichero 'filesystem.squashfs'. Primero debemos abrir y descomprimir el fichero en cuestión, y después dar permisos de escritura. Para poder trabajar con este tipo de fichero vamos a usar otra herramienta de código libre bajo licencia GPL-2.0 llamada *squashfs-tools*, que se puede consultar el código en Github [41]. Más concretamente la vamos a usar tanto para descomprimir como para volver a montar el fichero *squashfs* con los cambios pertinentes.

Finalmente una vez tenemos los ficheros que queremos ya modificados y guardados de forma que se pueda recrear la imagen ISO, vamos a usar otra vez la herramienta *xorriso*, con unos parámetros diferentes, para volver a comprimir todos los ficheros extraídos y poder obtener una imagen ISO con los cambios realizados.

6.3.2. Viabilidad de la solución

La viabilidad de la solución C se ha demostrado realizando varias pruebas, siguiendo los pasos descritos en la descripción de la solución y comprobando que el resultado era el deseado en cada caso. Han surgido algunas dificultades, sobre todo al inicio, para encontrar las herramientas adecuadas para nuestro propósito y también con el uso de la herramienta *xorriso* que puede llegar a ser un poco tedioso y complejo. También han aparecido algún contratiempo para entender bien como está estructurada una imagen ISO por dentro y como se puede manipular. Al final hemos podido llevar a cabo con éxito todos los pasos propuestos en la descripción, por lo tanto podemos determinar que la solución se puede llevar a la práctica. Incluso se ha implementado un prototipo funcional que serviría como demostración de su viabilidad tanto técnicamente como económicamente.

6.3.3. Aplicación de la metodología

Respecto al cálculo del potencial de mejora de escalabilidad de la solución, vamos a usar la metodología creada anteriormente en la sección 2.3 representándola en una tabla. Primero tenemos que tener en cuenta de que es una solución enfocada para la situación C, aunque al compararlo con todas las otras situaciones podremos ver si coincide el potencial de mejora con alguna de estas, aunque la solución inicialmente no está pensada para resolver otras situaciones.

Vamos a poner un valor sobre cada dimensión, empezando por la **dimensión A** que hace referencia al volumen de dispositivos que en este caso como la solución C no impacta sobre esta dimensión A, ya que se trata de modificar la imagen ISO que se usa para registrar ordenadores con el Workbench para poder generarla de una forma escalable, por eso va a tener un valor de mejora **Nulo** que equivale a cero puntos.

Para la **dimensión B** que sería el volumen de usuarios nuevos, en este caso el valor de mejora será **Alto**, es decir tres puntos, ya que para cada nuevo usuario tendríamos que generar una imagen ISO específica y con la solución C precisamente agiliza este proceso, por lo tanto va a permitir aumentar bastante la capacidad de nuevos usuarios usando la herramienta.

En la **dimensión C**, si nos preguntamos cuánto permite aumentar la solución propuesta la capacidad de registrar hardware distinto, la respuesta sería un valor **Medio**, es decir dos puntos, ya que tiene una repercusión considerable cuando se produce un error al registrar hardware nuevo y podemos generar de forma ágil una imagen ISO con los cambios del código para el usuario concreto y así para poder registrar el nuevo hardware correctamente.

Por último la **dimensión D** que está compuesta por tres subdimensiones, la **dimensión D1** referente al coste del registro por operario tendría un valor **Nulo** (cero puntos), ya que no afecta al coste que tiene el operario al realizar un registro. En cambio, la **dimensión D2** que es el coste del registro por soporte técnico, si nos preguntamos cuánto disminuirá el coste de mantenimiento en caso de implementar la solución C, la respuesta sería **Alto**, es decir tres puntos, ya que la solución propuesta reduce este coste significativamente. El ejemplo más claro es cuando hay un problema con una imagen ISO de un usuario específico y se tiene que rehacer, usando la solución C no supondría el mismo coste que actualmente. Y la **dimensión D3** referente al coste de infraestructura también disminuirá aunque no tan significativamente como en la subdimensión D2 pero si sobre todo con los costes de computación y distribución de las imágenes ISO. Es decir, necesitamos un ordenador potente para generar de cero una ISO y también espacio en la nube para guardar y poder enviar las imágenes ISO

a cada usuario. Si partimos desde una ISO genérica disminuimos estos costes, por lo tanto tendrá un valor de mejora **Medio** (dos puntos). Si hacemos la media redondeada de las subdimensiones D1, D2 y D3 nos sale un total de cinco puntos divididos entre tres subdimensiones redondeado tendría como media dos puntos que equivale a un valor **Medio** de mejora.

Una vez argumentado los valores asignados en cada dimensión para la solución C, vamos a mostrar una tabla general con el resultado del valor de mejora sobre cada dimensión para la situación C:

Potencial de mejora de la solución C en la situación C	
Dimensiones	Valor de mejora
<i>(Dim A) Volumen Dispositivos</i>	Nulo (0 puntos)
<i>(Dim B) Volumen Usuarios</i>	Alto (3 puntos)
<i>(Dim C) Diversidad Modelos</i>	Medio (2 puntos)
<i>(Dim D) Coste registro global</i>	Medio (2 puntos)
<i>(Dim D1) Coste registro operario</i>	Nulo (0 puntos)
<i>(Dim D2) Coste registro mantenimiento</i>	Alto (3 puntos)
<i>(Dim D3) Coste registro infraestructura</i>	Medio (2 puntos)
Puntuación Total (sobre 12)	7 de 12 puntos

Figura 9: Tabla Metodología (Situación C - Solución C)

6.3.4. Resultados

Referente a la puntuación de mejora de escalabilidad de **siete puntos sobre 12 para la solución C**. Hemos probado que es una solución viable técnicamente y económicamente haciendo una prueba de concepto. Esta prueba consiste principalmente en ejecutar unos comandos que nos permita modificar una imagen ISO y así demostrar la viabilidad de la solución y además da un punto de partida para que en caso de ser la solución escogida poder crear un prototipo. De momento se queda como la solución con mejor puntuación de mejora empatada con la solución B que también obtuvo siete puntos en total.

6.4. Solución D: Mostrar información sobre el error al usuario de forma categorizada y comprensible

6.4.1. Descripción de la solución

Partiendo del contexto de la situación D explicada anteriormente, vamos a proponer una solución basada en categorizar y mostrar información relevante al usuario para agilizar la posterior depuración del error. Al realizar la categorización vamos a poder acotar en cuál de las cuatro funciones principales del software Workbench, explicadas en la sección 1.2, se ha producido el fallo. Con esto obtenemos que el usuario que se encuentra un error de ejecución de la herramienta puede reportar los fallos de forma mucho más concisa y con una información mucho más valiosa para poder recrear el error rápidamente y así encontrar una solución al error.

6.4.2. Viabilidad de la solución

Para comprobar la viabilidad técnica de esta solución hemos hecho una prueba de concepto que consiste en tener un ordenador que sepamos de antemano que falla en una de las funciones concretas de la herramienta Workbench, más concretamente durante la obtención de información acerca del hardware del ordenador. Después realizamos una pequeña modificación en el código del Workbench en el mismo sitio donde se produce el fallo para que de una salida de error con un mensaje claro y comprensible para que el usuario sepa al menos en que fase se ha producido el error. Una vez hecho los cambios, vamos a comprobar que efectivamente al ejecutar la herramienta Workbench sobre el ordenador que falla, el mensaje de error es el deseado y realmente es comprensible y nos indica en que función se ha producido el fallo.

6.4.3. Aplicación de la metodología

Respecto al cálculo del potencial de mejora de escalabilidad de la solución, vamos a usar la metodología creada anteriormente en la sección 2.3 representándola en una tabla. Primero tenemos que tener en cuenta de que es una solución enfocada para la situación *D*. Es decir la solución inicialmente no está pensada para resolver otras situaciones, aunque al compararlo con las otras situaciones podremos ver si coincide el potencial de mejora con alguna de estas.

Vamos a poner un valor sobre cada dimensión, empezando por la **dimensión A** que hace referencia al volumen de dispositivos que en este caso no va a suponer ninguna mejora, es decir el impacto sobre la capacidad de registrar muchos ordenadores a la vez es **Nulo** que equivale a cero puntos, ya que la solución se centra en mejorar los mensajes de errores de ejecución de la herramienta Workbench.

Para la **dimensión B** que sería el volumen de usuarios, en este caso al plantearnos la pregunta de cuánto ha aumentado la capacidad de nuevos usuarios usando la herramienta, la respuesta sería un valor **Bajo** que equivale a uno punto, ya que no impacta de forma significativa sobre el volumen de usuarios, pero sí permite tener un volumen mayor de usuarios porque la solución nos va a permitir depurar de una forma ágil cuando ocurre un error de ejecución al usar la herramienta Workbench.

En la **dimensión C**, si nos preguntamos cuánto permite aumentar la solución propuesta la capacidad de registrar hardware distinto, la respuesta sería un valor **Medio**, es decir dos puntos, ya que la solución se enfoca en dar al usuario un mensaje de error más claro, que también repercute en la facilidad para depurar un error de ejecución y por lo tanto registrar hardware distinto siempre hay más probabilidad de errores, por eso si podemos agilizar la depuración de errores también podremos registrar hardware muy diferente.

Por último la **dimensión D** que está compuesta por tres subdimensiones, la **dimensión D1** referente al coste del registro por operario la solución va a permitir al usuario saber qué está fallando y poder arreglarlo o reportarlo rápidamente, por lo que tendría un valor de mejora **Alto** (tres puntos). Una cosa muy similar a lo expuesto antes es lo que tenemos en la **dimensión D2** que es el coste del registro por soporte técnico, en este caso el valor de mejora será **Alto** (tres puntos) porque la solución está enfocada a mostrar mensajes de error más comprensibles y por lo tanto también a la depuración en caso de error al ejecutar la herramienta Workbench. Esto impacta directamente con el coste del soporte técnico que es el encargado de realizar dicha depuración. Y la **dimensión D3** referente al coste de infraestructura o reducción de los materiales requeridos para usar la herramienta podemos responder que la solución tendrá un valor de mejora **Bajo**, es decir uno punto, ya que en este caso vamos a tener

prácticamente los mismos costes para usar la herramienta. Si hacemos la media redondeada de las subdimensiones D1, D2 y D3 nos sale un total de siete puntos divididos entre tres subdimensiones redondeado tendría como media tres puntos que equivale a un valor **Medio** de mejora.

Una vez argumentado los valores asignados en cada dimensión para la solución D, vamos a mostrar una tabla general con el resultado del valor de mejora sobre cada dimensión para la situación D:

Potencial de mejora de la solución D en la situación D	
Dimensiones	Valor de mejora
<i>(Dim A) Volumen Dispositivos</i>	Nulo (0 puntos)
<i>(Dim B) Volumen Usuarios</i>	Bajo (1 punto)
<i>(Dim C) Diversidad Modelos</i>	Medio (2 puntos)
<i>(Dim D) Coste registro global</i>	Alto (3 puntos)
<i>(Dim D1) Coste registro operario</i>	Alto (3 puntos)
<i>(Dim D2) Coste registro mantenimiento</i>	Alto (3 puntos)
<i>(Dim D3) Coste registro infraestructura</i>	Bajo (1 punto)
Puntuación Total (sobre 12)	6 de 12 puntos

Figura 10: *Tabla Metodología (Situación D - Solución D)*

6.4.4. Resultados

Como resultado de esta solución tenemos una puntuación de mejora de escalabilidad de **seis puntos sobre 12 para la solución D**. Hemos probado que es una solución viable técnicamente y económicamente, ya que hemos realizado una prueba de concepto satisfactoriamente.

A falta de ver la puntuación de la solución E, vamos a decir que no es realmente una solución que aporte una mejora de escalabilidad significativa porque se encuentra justo en la línea de los seis puntos sobre 12 que sería obtener la mitad de todos los puntos posibles. Como no esta por encima de esta línea que nos hemos marcado como aportación significativa de mejora de escalabilidad. Difícilmente esta solución será escogida como prioritaria.

6.5. Solución E: Crear ordenadores virtualizados para aumentar la compatibilidad con hardware diferente

6.5.1. Descripción de la solución

Partiendo del contexto de la situación E explicada anteriormente, vamos a proponer una solución basada en la virtualización de ordenadores. Es decir, cuando se quiera mejorar la compatibilidad con ordenadores concretos, con una arquitectura o hardware diferente de los modelos compatibles hasta la fecha.

Vamos a buscar una herramienta, preferiblemente de código abierto, que nos permita crear un ordenador virtual con un hardware específico que nosotros podamos predefinir. Después vamos a ver cómo de una forma sencilla y rápida podemos ejecutar nuestra herramienta Workbench sobre este ordenador virtual previamente creado.

Después de una búsqueda por Internet sobre las alternativas de código libre existentes enfocadas a la creación de máquinas virtuales y emulación de hardware, encontramos **libvirt** [42], una API de código abierto muy extendida dentro de las herramientas de gestión para administrar la virtualización de hardware. Más concretamente para generar ordenadores virtuales vamos a usar una herramienta llamada QEMU [43] (abreviado Quick EMUlator) bajo licencia GPLv2, capaz de cumplir con los objetivos planteados y aunque no puede emular todos los modelos o las arquitecturas existentes, sí que soporta una larga lista de dispositivos. Con esta herramienta de código libre seremos capaces de emular hardware distinto usando una misma máquina como host, permitiendo así obtener hardware muy diferente de forma rápida y con muy pocos recursos en comparación a obtener el hardware físicamente.

Una vez aclaradas todas las opciones y parámetros importantes que nos ofrece la herramienta, se crean dos máquinas totalmente distintas y se registran con el hardware virtual para comprobar si la herramienta Workbench es compatible con ese hardware concreto y así aumentar su compatibilidad. Al final vamos a obtener un mecanismo que nos permite validar que la herramienta es compatible con un modelo específico de ordenador, es decir un hardware concreto.

6.5.2. Viabilidad de la solución

La viabilidad de la solución *E* se ha demostrado realizando varias pruebas, siguiendo los pasos descritos en la descripción de la solución y comprobando que el resultado era el deseado en cada caso. Han surgido algunas dificultades al inicio con el uso de la herramienta QEMU, ya que te da tantísimas opciones y parámetros que se pueden usar, que no ha sido trivial encontrar los parámetros concretos que nos interesan para la emulación del hardware. Una vez hemos conseguido tener esta primera máquina con los parámetros del hardware concreto que queremos emular, hemos demostrado que la solución es viable y para generar la segunda máquina solo ha sido cuestión de ir variando los valores de estos parámetros.

6.5.3. Aplicación de la metodología

Respecto al cálculo del potencial de mejora de escalabilidad de la solución *E*, vamos a usar la metodología creada anteriormente en la sección 2.3 representándola en una tabla. Primero tenemos que tener claro que es una solución enfocada para la situación *E*.

Vamos a poner un valor sobre cada dimensión, empezando por la **dimensión A** que hace referencia al volumen de dispositivos que en este caso como la solución *E* se centra en generar ordenadores virtuales con hardware concreto, no aumenta la capacidad de registrar muchos ordenadores a la vez. Por lo tanto el impacto sobre la dimensión A es **Nulo** que equivale a cero puntos. Sin embargo sí nos permite validar que la herramienta es compatible con un modelo de ordenador (hardware concreto)

Para la **dimensión B** que sería el volumen de usuarios, en este caso al plantearnos la pregunta pertinente, la respuesta sería un valor **Nulo** que equivale a cero puntos, ya que no impacta sobre el volumen de usuarios nuevos sino sobre el volumen de ordenadores con hardware distinto.

En la **dimensión C**, si nos preguntamos cuanto permite aumentar la solución propuesta la capacidad de registrar hardware distinto, la respuesta sería un valor **Alto**, es decir tres puntos, ya que la solución permite entrenar la herramienta Workbench usando ordenadores virtuales precisamente para aumentar el número de modelos diferentes compatibles.

Por último la **dimensión D** que está compuesta por tres subdimensiones, la **dimensión D1** referente al coste del registro por operario tendría un valor **Nulo** (cero puntos), ya que la solución no influye sobre el coste del registro por operario. En cambio, la **dimensión D2** que es el coste del registro por soporte técnico, en este caso el valor de mejora será **Alto** (tres puntos) porque la solución está enfocada a poder entrenar la herramienta y comprobar si un hardware concreto es compatible con la herramienta. Esto impacta fuertemente con el coste del soporte técnico, ya que es el encargado de realizar el mantenimiento de la herramienta y asegurar que funciona correctamente. Respecto a la **dimensión D3** referente al coste de infraestructura o reducción de los materiales requeridos para usar la herramienta, podemos responder que la solución tendrá un valor de mejora **Medio**, es decir dos puntos, ya que es este caso para usar la herramienta necesitar tener un ordenador donde usarla, si físicamente no se puede conseguir, vamos a reducir el coste de los recursos si los podemos crear virtualmente. Si hacemos la media redondeada de las subdimensiones D1, D2 y D3 nos sale un total de cinco puntos divididos entre tres subdimensiones redondeado tendría como media dos puntos que equivale a un valor **Medio** de mejora.

Una vez argumentado los valores asignados en cada dimensión para la solución *E*, vamos a mostrar una tabla general con el resultado del valor de mejora sobre cada dimensión para la situación *E*:

Potencial de mejora de la solución E en la situación E	
Dimensiones	Valor de mejora
<i>(Dim A) Volumen Dispositivos</i>	Nulo (0 puntos)
<i>(Dim B) Volumen Usuarios</i>	Nulo (0 puntos)
<i>(Dim C) Diversidad Modelos</i>	Alto (3 puntos)
<i>(Dim D) Coste registro global</i>	Medio (2 puntos)
<i>(Dim D1) Coste registro operario</i>	Nulo (0 puntos)
<i>(Dim D2) Coste registro mantenimiento</i>	Alto (3 puntos)
<i>(Dim D3) Coste registro infraestructura</i>	Medio (2 puntos)
Puntuación Total (sobre 12)	5 de 12 puntos

Figura 11: *Tabla Metodología (Situación E - Solución E)*

6.5.4. Resultados

Como resultados de esta solución tenemos una puntuación de mejora de escalabilidad de **cinco puntos sobre 12 para la solución E**. Hemos probado que es una solución viable técnicamente y económicamente, ya que hemos creado dos ordenadores con hardware virtual diferente y los hemos registrado usando la herramienta Workbench. Podemos afirmar sabiendo ya las puntuaciones de las otras soluciones que esta solución no será escogida como prioritaria y que solo sería deseable implementarla en caso de enfocarnos solo sobre la dimensión C referente a la capacidad de adaptar el software para registrar ordenadores diferentes con hardware muy distinto.

7. Análisis de la escalabilidad de las soluciones validadas

Como se puede observar a continuación, la Figura 12 representa un resumen de las puntuaciones obtenidas en cada dimensión y su puntuación total después de aplicar la metodología creada sobre cada una de las soluciones validadas.

Potencial de mejora de las soluciones propuestas					
Dimensiones	Solución A	Solución B	Solución C	Solución D	Solución E
(Dim A) Volumen dispositivos	Alto (3 puntos)	Bajo (1 punto)	Nulo (0 puntos)	Nulo (0 puntos)	Nulo (0 puntos)
(Dim B) Volumen usuarios	Nulo (0 puntos)	Bajo (1 punto)	Alto (3 puntos)	Bajo (1 punto)	Nulo (0 puntos)
(Dim C) Diversidad modelos	Nulo (0 puntos)	Alto (3 puntos)	Medio (2 puntos)	Medio (2 puntos)	Alto (3 puntos)
(Dim D) Coste registro global	Bajo (1 punto)	Medio (2 puntos)	Medio (2 puntos)	Alto (3 puntos)	Medio (2 puntos)
(Dim D1) Coste registro operario	Medio (2 puntos)	Medio (2 puntos)	Nulo (0 puntos)	Alto (3 puntos)	Nulo (0 puntos)
(Dim D2) Coste registro mantenimiento	Nulo (0 puntos)	Alto (3 puntos)	Alto (3 puntos)	Alto (3 puntos)	Alto (3 puntos)
(Dim D3) Coste registro infraestructura	Nulo (0 puntos)	Nulo (0 puntos)	Medio (2 puntos)	Bajo (1 punto)	Medio (2 puntos)
Puntuación Total	4 de 12 puntos	7 de 12 puntos	7 de 12 puntos	6 de 12 puntos	5 de 12 puntos

Figura 12: Tabla Metodología Soluciones

Vamos a repasar por orden, empezando por la solución con mayor potencial de mejora de escalabilidad, tanto las situaciones evaluadas como las puntuaciones de sus respectivas soluciones.

En primera posición nos encontramos con un empate teórico entre la **solución B y C** que han obtenido un valor de mejora de **siete puntos** cada una. Recordar que estas soluciones salen después de evaluar la situación B y C respectivamente. La situación C nos plantea el contexto de tener un volumen considerable de nuevos usuarios dentro del sistema, que quieren usar la herramienta Workbench y subir la información del ordenador al inventario Devicehub. Para esta situación se ha propuesto modificar la imagen ISO genérica del Workbench para cada usuario específico sin crear una nueva imagen ISO desde cero, si no extrayendo la imagen ISO y modificando los datos del usuario directamente sin generar una imagen ISO de cero. La situación B nos plantea el caso de tener un volumen pequeño de ordenadores pero muy diferentes entre sí, con modelos y fabricantes muy diversos pendientes de registrar por la herramienta Workbench. En este caso se propuso un sistema de depuración para la mejora continua de la herramienta y poder registrar ordenadores nuevos para la herramienta. De momento vamos a dejarlo como un empate y a seguir con las demás posiciones. Una vez analizadas todas las otras puntuaciones vamos a realizar el desempate profundizando en las puntuaciones de estas dos soluciones, tal como se puede apreciar en la sección 7.1.

En segunda posición está la **solución D** que surge de la necesidad de mejorar la situación D que nos plantea el caso de un ordenador concreto que falla durante la ejecución del Workbench, antes de subir la información del ordenador al inventario Devicehub. Partiendo de este contexto surge una propuesta de solución, que posteriormente se valida, que consiste en mostrar información relevante y comprensible sobre el error al usuario. Esta solución tiene una puntuación de **seis puntos** sobre 12 posibles después de obtener un valor Alto en media sobre la dimensión D referente al coste del registro en global. Partiendo de esta premisa podemos afirmar que implementar esta solución aportaría una gran mejora de escalabilidad respecto el coste de registro, sobre todo el coste registro del operario y del mantenimiento o soporte técnico. También destacar que tiene un valor Medio en la dimensión

C sobre la capacidad de adaptar el software para registrar ordenadores con modelos distintos.

En tercera posición se encuentra la **solución E** que es fruto de dar respuesta a la situación E nos plantea el caso de tener muy pocos ordenadores para el testing con hardware diverso y queremos probar de registrar ordenadores muy diferentes con hardware variado, incluso con arquitecturas diferentes, sin tener que obtener y montar físicamente todos estos modelos concretos de ordenadores. Para este contexto se propone una solución basada en crear ordenadores virtualizados para aumentar la compatibilidad con hardware diferente. Esta solución E obtiene un total de **cinco puntos** desglosados en un valor Alto para la dimensión C referente a la diversidad de modelos y un valor Medio sobre la dimensión D referente al coste de registro. Sobre las otras dimensiones tiene un valor Nulo porque representa que no aporta ninguna mejora sobre estas dimensiones.

En la última posición tenemos la **solución A** que nace de la situación A que nos plantea tener un gran volumen de ordenadores similares o iguales pendientes de registrar por la herramienta Workbench. Para esta situación surge una propuesta de solución que consiste en distribuir la imagen ISO del Workbench por red usando un servidor PXE. Esta solución tiene la peor puntuación con **cuatro puntos** y por eso está el último de la lista. Solo destaca sobre la dimensión A, enfocada a la capacidad de registrar muchos dispositivos a la vez, que tiene un valor de mejora Alto. Sobre las otras dimensiones tiene un valor Nulo o Bajo porque representa que no aporta una mejora realmente significativa sobre estas dimensiones.

7.1. Lista priorizada de las soluciones

A continuación la lista priorizada de las soluciones propuestas ordenada por mayor mejora de escalabilidad aportada:

Lista priorizada de las soluciones propuestas	
Posición soluciones	Puntuación total (sobre 12)
1ª – Solución C	7 puntos
2ª – Solución B	7 puntos
3ª – Solución D	6 puntos
4ª – Solución E	5 puntos
5ª – Solución A	4 puntos

Figura 13: *Lista priorizada de las soluciones*

Según podemos observar en la Figura 13, aparecen las soluciones en orden de implementación deseado teniendo en cuenta todas las dimensiones definidas en la metodología creada al inicio de este trabajo. Aunque técnicamente hay un empate por la primera posición de la lista, se ha escogido como **prioritaria la solución C por encima de la solución B por dos motivos** principalmente que vamos argumentar a continuación.

El primer motivo sería **el tipo de usuario que usa el Workbench** actualmente, el proyecto de eReuse se centra en reacondicionadoras pequeñas que procesan poco volumen de dispositivos y que además suelen ser dispositivos de fabricantes y modelos diferentes. Si queremos aumentar el número de dispositivos a registrar al mes, sabiendo el tipo de usuarios que se enfoca la herramienta Workbench, vamos a necesitar que la solución que tenga una muy buena puntuación en la dimensión B, referente a la capacidad de dar de alta un volumen considerable de nuevos usuarios usando la herramienta sin aumentar los costes. Así aunque cada usuario o reacondicionadora registre un volumen pequeño de dispositivos, si doblamos el número de usuarios efectivamente vamos a registrar el doble de dispositivos al mes. Por eso motivo la solución escogida como prioritaria es la solución C.

También hay un segundo motivo que se basa en **la distribución de la puntuación obtenida**. Es decir, que la solución C es la más prioritaria porque solo tiene una dimensión con un valor de mejora Nulo o Bajo, en cambio la solución B no aporta una mejora representativa ni sobre la dimensión A ni sobre la dimensión B.

Por este hecho, aunque las dos soluciones hayan obtenido la misma puntuación total, si entramos en detalles después de exponer los dos motivos anteriores, podemos concluir claramente que **la solución C es más prioritaria y deseable de implementar primero**.

8. Prototipo de la solución escogida

Primero vamos a evaluar en que nivel de madurez tecnológica, TLR se encuentra cada solución validada anteriormente antes de entrar en detalles de como se ha implementado prototipo.

Empezando por la solución *A* que es la solución con un nivel de madurez tecnológica más bajo, aunque para validar técnicamente la solución hicimos una prueba de concepto explicada en la sección 6.1.2, partiendo del precedente que ya teníamos implementado anteriormente sobre la situación *A*. Esta prueba de concepto deja el nivel de madurez tecnológica de la **solución *A* en un TRL de nivel 3**.

El nivel de madurez de la solución *B* es alto, ya que era la solución candidata juntamente con la solución *C* y se llegó a un nivel de madurez tecnológica, es decir la **solución *B* tiene un TRL de nivel 5**, ya que se ha validado la tecnología en un entorno relevante como puede ser un caso real como se expone en la sección 6.2.2.

Igual que en la solución anterior, la solución *C* tendrá un nivel de madurez alto y aun más sabiendo que es la solución escogida como prioritaria y por lo tanto vamos a implementar un prototipo de esta solución. Y tal como se describe en la sección 6.3.2 al tener una prototipo funcional con el cual podemos hacer una demostración. La **solución *C* va a tener un TRL de nivel 7**.

La solución *D* se encuentra en un nivel de madurez tecnológica bajo, ya que se realizó una prueba de concepto, tal como se explica en la sección 6.4.2, pero no se llegó a validar la tecnología usada. Por lo tanto la **solución *D* tiene un TRL de nivel 3**.

Por último la solución *E* se hizo varias pruebas aparte de una prueba experimental de concepto tal como se expone en la sección 6.5.2, por lo que la **solución *E* tiene un TRL de nivel 4**.

Después de evaluar el nivel de madurez tecnológica de cada solución vamos a detallar en qué consiste el prototipo funcional de la solución escogida como prioritaria que en este caso sería la solución *C*, para poder recrear y usar el prototipo para demostrar que realmente es funcional.

8.1. Descripción del prototipo funcional

Este prototipo parte siempre de una imagen ISO genérica del Workbench como base y consiste en un conjunto de comandos que generan una nueva imagen ISO modificada para un usuario específico. Este conjunto de comandos se empaquetan en un script usando lenguaje Bash para automatizar todos los pasos posibles descritos anteriormente durante la descripción de la solución *C*. Dentro del script vamos a usar principalmente dos programas de línea de comandos de código libre llamados *xorriso* y *squashfs-tools* explicados en la sección 6.3.

El prototipo de la solución *C* se compone de **cinco partes** bien diferenciadas que se detallan a continuación.

La primera parte se centra en **preparar y configurar todo el entorno**, juntamente con la instalación de los paquetes de software o dependencias necesarias para que el prototipo funcione correctamente. En este punto se crearan las variables de entorno necesarias, carpetas temporales y también se instalaran los paquetes *xorriso* y *squashfs-tools*.

La segunda parte consiste en **abrir y extraer la imagen ISO usando la herramienta *xorriso***, posteriormente abrir y extraer el fichero comprimido 'filesystem.squashfs' contenido dentro de la carpeta 'live' de la imagen ISO descomprimida usando la herramienta *squashfs-tools*.

La tercera parte consiste en **modificar y guardar los cambios pertinentes que serían las credenciales del usuario** que en este caso se encuentran dentro del fichero 'filesystem.squashfs'. Para realizar la modificación podríamos usar cualquier editor de texto por línea de comandos. En nuestro caso hemos usado *nano* por comodidad y familiarización con este editor. Para volver a comprimir el fichero 'filesystem.squashfs' hemos usado otra vez la herramienta *squashfs-tools*.

La cuarta parte consiste en **reconstruir la imagen ISO conservando los cambios hechos** dentro de ella. Para eso vamos a usar la herramienta *xorriso* otra vez pero con unos parámetros diferentes a la segunda parte.

La quinta parte se trata de que el **script tenga la propiedad de idempotencia**, es decir que por muchas veces que se ejecute el script siempre se cumpla que el estado del sistema sigue siendo el mismo.

Por último la sexta parte aunque no es obligatoria su ejecución para conseguir los resultados esperados por la solución, si es muy recomendable porque se asegura que los cambios hechos sobre la imagen ISO se han realizado correctamente. Esta parte se centra en realizar un **test sobre la imagen ISO modificada** que consiste en arrancar la imagen ISO en un ordenador virtualizado usando la herramienta QEMU, también usada en la solución E explicada en la sección 6.5.

El prototipo está publicado en el Github de eReuse, más concretamente dentro de la herramienta Workbench Live, ya que se trata de una mejora sobre las imágenes ISO que genera precisamente la herramienta Workbench Live. De momento se encuentra en la rama *devel*, ya que solo es un prototipo funcional, se puede consultar el código en el Github [44].

8.2. Resultados usando el prototipo funcional

Para exponer un poco la mejora real que aporta la implementación del prototipo funcional, primero vamos a exponer de donde partimos antes de usar el prototipo. Actualmente, cuando un nuevo usuario requiere una imagen ISO específica para poder usar la herramienta Workbench, se le tenía que generar una imagen ISO específica prácticamente desde cero, te puedes ahorrar la descarga de paquetes necesarios para generar dicha ISO, que también suponen un tiempo que puede llegar a ser considerable dependiendo de la calidad que tengamos de conexión a Internet. Para las pruebas no vamos a tener en cuenta este paso, igual que en el prototipo tampoco vamos a computar el tiempo que tarda a descargar e instalar los paquetes requeridos para el correcto funcionamiento del prototipo.

Vamos a explicar las pruebas realizadas para poder medir un poco la mejora real que nos ha implicado usar el prototipo funcional desarrollado durante este trabajo. Como punto de partida y **entorno vamos a usar siempre el mismo ordenador**, concretamente uno de gama alta, con unas prestaciones elevadas como son un procesador de 8 hilos que puede trabajar a 4,40 GHz con 16 GB de memoria RAM. Añadir que los tiempos expuestos más adelante son orientativos y que se han obtenido realizando la media de tres mediciones, así podemos obtener y comparar los resultados de una forma más rigurosa.

Para ir entrando en detalles, sin usar el prototipo el tiempo de ejecución medio que tarda en generarse una imagen ISO desde cero usando el entorno expuesto en el párrafo anterior es aproximadamente

de alrededor de unos **16 minutos** con el ordenador dedicado exclusivamente para esta tarea. Otra medición que hemos hecho fuera del entorno para tener una orientación de cuánto tiempo puede tardar el mismo proceso usando un ordenador menos potente y los resultados han sido unos tiempos alrededor de una hora. Después hemos usado el prototipo funcional para generar una nueva ISO modificada a partir de una imagen ISO ya creada previamente. Ejecutar el prototipo, sin contar el tiempo de descargar e instalación de los paquetes requeridos, ha tardado en media un tiempo aproximado de **4 minutos** en generar la nueva imagen ISO modificada en el entorno predefinido con el ordenador de gama alta. Después hemos probado de ejecutar el prototipo usando un ordenador menos potente, igual que el caso anterior, y los tiempos también se han incrementado bastante con unos tiempos alrededor de 10 minutos aproximadamente.

Al final como conclusión del análisis podemos observar que la implementación del prototipo nos reduce en media en cuatro veces el tiempo que necesitábamos antes para proveer una imagen ISO específica a un usuario. Es decir que ahora la herramienta Workbench es más escalable al menos sobre la situación en la que se basa la solución C escogida y que hace referencia a tener un volumen considerable de nuevos usuarios pendientes de alta dentro del sistema y que quieren usar la herramienta lo antes posible. Por lo que podríamos afirmar que la capacidad de tener un gran número de usuarios nuevos usando la herramienta se ha cuadruplicado sin aumentar los costes, ya que usamos el mismo entorno, en este caso el mismo ordenador.

Por último vamos a concluir con un ejemplo práctico real usando los pronósticos esperados desde eReuse. Si actualmente hay 10 usuarios activos que usan unas 10 imágenes ISO diferentes, y se espera triplicar el número de usuario en un mes. Es decir que tendremos 30 nuevos usuarios que van a necesitar unas 10 imágenes ISO cada uno, en total 300 imágenes ISO al mes. Antes para afrontar este volumen de 30 nuevos usuarios de la herramienta Workbench tardaba aproximadamente, usando un ordenador potente, unos 16 minutos por 300 imágenes ISO, es decir unos **4800 minutos al mes** que serían unas 80 horas mensuales solo para generar las imágenes ISO. En cambio, ahora usando el prototipo funcional se pueden generar las 300 mismas imágenes ISO con un tiempo aproximado de unos **1200 minutos** en total, es decir unas 20 horas aproximadamente.

Como conclusiones podemos afirmar que realmente el prototipo aporta una mejora de escalabilidad, que aunque **reduce un 75% el tiempo de generación de las imágenes ISO**, sigue subiendo el coste de forma lineal respecto el número de nuevos usuarios usando el prototipo implementado durante este trabajo. Después de llegar a esta conclusión se propuso una variante o ampliación de la solución C, que se podría usar como ampliación para un futuro trabajo, que consiste en agregar una partición persistente dentro de una memoria USB donde montamos la imagen ISO del Workbench. Con esta partición persistente se podría desacoplar las credenciales del usuario y también la configuración de ejecución del Workbench, poniendo toda esta información en un fichero aparte dentro de la partición persistente donde se quedaría guardada. Así podríamos obtener una imagen ISO genérica del Workbench reusable para todos los usuarios y configuraciones posibles. Como valor extra añadir que sería posible guardar en local dentro la memoria USB y de forma persistente los datos del registro, también llamados *snapshots*, en caso de no tener conexión a Internet y así poder subirlos manualmente al Devicehub en otro momento cuando si se tenga conexión a Internet. Con esta ampliación de la solución C podríamos realmente escalar en volumen de usuarios sin aumentar el coste linealmente, ya que solo necesitaríamos una sola imagen para todos los usuarios, y sería el propio usuario el encargado de poner sus datos y la configuración del Workbench deseada dentro del fichero correspondiente. Por lo tanto el tiempo que nos costaría dar de alta cualquier número de nuevos usuarios sería siempre los 4 minutos que tarda el prototipo en generar una nueva imagen ISO.

8.3. Futura integración del prototipo

La integración del prototipo dentro del ecosistema de eReuse es una tarea que queda fuera del alcance de este trabajo por la complejidad que supone la integración con los diferentes sistemas que intervienen y el tiempo limitado que tenemos para la realización de este trabajo.

Una posible futura integración dentro del ecosistema de eReuse, sería dar la posibilidad al usuario directamente de descargar la imagen ISO asociada a su usuario (personalizada) desde la plataforma web Devicehub partiendo de una imagen ISO genérica del Workbench. Para lograr esto vamos a usar como punto de partida el prototipo implementado durante este trabajo pero controlada por el servidor. Una posible implementación sería ejecutar el prototipo cada vez que se dé de alta un nuevo usuario y genere una imagen ISO modificada para dicho usuario. El prototipo actualmente solo genera una nueva ISO modificada con las credenciales de dicho usuario partiendo de una imagen ISO genérica. Por lo que faltaría realizar todo un conjunto de desarrollos que implican desde subir el prototipo a los servidores de eReuse, implementar una pequeña interfaz dentro del Devicehub, como podría ser un botón, donde el usuario pueda descargar la imagen ISO y también vincular la creación de cuentas con la ejecución automática del prototipo y también otros desarrollos menores que no vamos a mencionar por su poca relevancia.

9. Sostenibilidad y compromiso social

9.1. Autoevaluación

En referencia al ámbito de la sostenibilidad, decir que ya conocía el concepto que se le da hoy en día, enfocado en las tres dimensiones que vamos a analizar más adelante [45]. Tampoco considero que tenga unos conocimientos profundos en este ámbito concreto para poder opinar con un criterio rotundo. Pero sí que tengo algo de experiencia dentro del mundo de la economía circular y más concretamente la reutilización, por mi trabajo de voluntariado en eReuse durante varios años. También remarcar que durante la carrera nos han hecho énfasis sobre este tema en algunas asignaturas, sobre todo como competencias transversales o realizando algún trabajo muy concreto en alguna asignatura.

Realmente no se le da toda la importancia que requiere actualmente. Podríamos definir, desde mi punto de vista, el primer y principal punto de la sostenibilidad es asumir que la naturaleza y el medio ambiente no son una fuente inagotable de recursos, siendo necesario su uso racional y de forma eficiente, creativa y óptima. También implica promover un desarrollo económico que genere riqueza equitativa para todos sin dañar el medio ambiente. Una forma de conseguirlo es manteniendo recursos en su máximo valor y evitar la generación de residuos, que es el principio esencial de la economía circular [7]. Por último, también hay que potenciar el desarrollo social buscando la cohesión entre comunidades y culturas diferentes. Resumiendo así, las tres dimensiones que a continuación vamos a poner más énfasis: la ambiental, la económica y la social.

Como conclusión, reafirmar que un buen nivel de sostenibilidad a todos los niveles es algo necesario y básico para enfrentarse a retos globales en uno mundo lleno de incertidumbre, volatilidad y complejidad.

9.1.1. Dimensión ambiental

- **¿Has estimado el impacto ambiental que tendrá la realización del proyecto?**

No de una forma exacta, pero si podemos afirmar que tendrá un impacto ambiental destacado, ya que el proyecto eReuse.org se enfoca en potenciar la reutilización de dispositivos electrónicos desde l'economía circular. Si mejoramos la herramienta que facilita este objetivo, estamos generando un impacto ambiental.

- **¿Te has planteado minimizar el impacto, por ejemplo, reutilizando recursos?**

Sí, de hecho la herramienta que vamos a mejorar y en la que se basa el proyecto tiene la finalidad de registrar ordenadores para poder realizar una reutilización de ellos. Dentro del apartado del presupuesto 4.1.2 ya mencionamos el hecho de reutilizar recursos materiales, como por ejemplo ordenadores de pruebas para usar la herramienta Workbench.

- **¿Has cuantificado el impacto ambiental de la realización del proyecto? ¿Qué medidas has tomado para reducir el impacto?**

No de una forma detalla ni explícita, pero si durante toda la realización del proyecto se han ido tomando medidas para reducir el impacto en todo lo posible, como usar ordenadores de

segunda mano para probar la herramienta o mirar de reducir al máximo los recursos materiales necesarios para la realización del proyecto. Al final después de implementar el prototipo como se reduce en cuatro veces el tiempo de computación necesario lo que genera un impacto proporcional de reducción del CO₂ en cuatro veces también.

9.1.2. Dimensión económica

- **¿Has estimado el coste de la realización del proyecto (recursos humanos y materiales)?**

No de una forma exhaustiva, pero sí con relación a los recursos materiales y teniendo en cuenta el aspecto de sostenibilidad, ya que es un tema que siempre me ha motivado bastante. Sin embargo, en el aspecto de los recursos humanos no había estimado los costes.

- **¿Has cuantificado el coste (recursos humanos y materiales) de la realización del proyecto? ¿Qué decisiones has tomado para reducir el impacto?**

No se ha cuantificado de una forma detallada aunque si se ha realizado una aproximación de los costes, expuestos en la sección 4. Se ha reducido el número de hora de dirección y consultoría externa para ajustar al máximo el impacto económico. También gracias al prototipo implementado se han reducido cuatro veces las horas de soporte técnico necesario para cada alta nueva de usuario.

9.1.3. Dimensión social

- **¿Qué crees que te va a aportar a nivel personal la realización de este proyecto?**

Aportar mi granito de arena y contribuir al mundo sostenible que me gustaría ver algún día, también para las futuras generaciones. Y a nivel interno supongo que satisfacción personal por realizar una aportación positiva en un proyecto sostenible y que a la vez promueva valores tanto de código abierto como de reutilización de recursos, que son dos ámbitos en los cuales creo y quiero dedicar mi tiempo.

- **¿La realización de este proyecto ha implicado reflexiones significativas a nivel personal, profesional o ético de las personas que han intervenido?**

Si, que realmente estamos haciendo muy poca cosa para cambiar la situación actual de consumo y que se pueden usar las tecnologías de una forma mucho más responsable.

10. Conclusiones y futuro trabajo

Vamos a empezar las conclusiones exponiendo cómo hemos superado los distintos objetivos marcados al inicio del trabajo y explicados en la sección 2.1. Después vamos a reflexionar sobre los aprendizajes personales adquiridos durante la realización de este trabajo. Por último se explicará el trabajo futuro que se quedó pendiente para llevar a cabo en otro trabajo o como ampliación de este mismo.

10.1. Objetivos cumplidos

Empezando por el objetivo inicial, que consistía en crear una metodología para valorar cuantitativamente la mejora de escalabilidad de una solución sobre un software de registro de ordenadores, tal como se describe en la sección 2.3.

Después exponer que tanto el segundo como el tercer objetivo también han sido cumplidos, ya que hemos evaluado las diferentes situaciones que nos han planteado desde el equipo de eReuse, tal como se explica en la sección 5 y también sus posibles soluciones juntamente con la aplicación de la metodología creada sobre cada solución validada técnicamente y económicamente, como se puede apreciar en la sección 6.

Por último el objetivo final del trabajo también ha sido alcanzado, ya que consistía en hacer una priorización por mayor grado de escalabilidad, tal como se observa en la Figura 13, para poder escoger una solución como prioritaria y crear un prototipo funcional de esta solución escogida, tal como se expone en la sección 8.

Por tanto, podemos decir que hemos conseguido avanzar en mejorar la madurez tecnológica de la implementación de todas las soluciones, especialmente en las que se han valorado como prioritarias.

10.2. Conclusiones personales

Como conclusiones personales, después de realizar este trabajo sobre las claves para escalar un software como el Workbench te das cuenta de que, aunque parece relativamente fácil realizar una valoración de mejora sobre un desarrollo pendiente, realmente puede ser complejo evaluar de una forma rigurosa la mejora de escalabilidad que nos puede aportar un desarrollo pendiente.

Desde el equipo de eReuse se han ido encontrando con ciertas situaciones que impiden que la herramienta Workbench sea escalable y realmente se pueda distribuir y crecer su uso sin aumentar los costes. Como existían varias situaciones que eran deseables y prioritarias de encontrar una solución escalable, pero solo teníamos recursos para desarrollar una solución, teníamos que poder priorizar y escoger una solución que nos aportara una mayor mejora respecto las otras.

Finalmente llegamos a la conclusión que realmente el objetivo general de este tipo de herramientas es registrar ordenadores. Así podremos aumentar el número de dispositivos que se registran cada mes. Nos ha quedado claro que el tipo de usuario de la herramienta específica del Workbench son entidades reacondicionadoras que registran un volumen pequeño de dispositivos, normalmente de fabricantes y modelos diferentes. Llegamos a la conclusión que para aumentar el número de dispositivos que se registran cada mes, necesitábamos poder aumentar el número de nuevos usuarios

usando la herramienta. Por lo que se terminó escogiendo la solución C que daba una mayor mejora de escalabilidad sobre la dimensión B que hace referencia al volumen de usuarios. Finalmente tal como se explica en la sección 8.2, gracias al prototipo hemos podido demostrar que es una solución escalable aunque mejorable, habría sido muy deseable poder realizar la ampliación de la solución C y también poder integrar completamente la solución dentro del ecosistema de eReuse.

10.3. Trabajo futuro

Queda pendiente terminar el desarrollo del prototipo para poder integrarlo dentro del ecosistema de eReuse, dado que el tiempo del TFG no cubre el tiempo necesario para implementar el extenso esfuerzo que implica este trabajo.

Como ampliación se podrían evaluar muchas más situaciones y mejorar la metodología creada añadiendo nuevas dimensiones sobre las que sea importante que el software tenga la capacidad de adaptarse, es decir ser escalable. También se podría aumentar la granularidad de los valores de mejora para obtener una precisión mayor sobre las puntuaciones.

Otra ampliación futura a la solución escogida por este trabajo sería, partiendo del prototipo ya implementado, replicar el mismo proceso pero en vez de modificar los datos referentes a las credenciales del usuario, que se puedan modificar los parámetros de ejecución del Workbench para que el usuario pueda generar una imagen ISO con una configuración específica. Es decir que el Workbench realice el borrado de datos o las pruebas de diagnóstico dependiendo de los requisitos que tenga el usuario final sobre los dispositivos a procesar por el Workbench.

Por último dejar constancia que a base de ir iterando sobre la metodología creada, proponiendo nuevas soluciones e integrando las soluciones validadas, se podría terminar obteniendo, en un futuro, un software maduro y escalable que pueda estar en producción y que los usuarios lo puedan usar como una herramienta estable y mantenible.

Referencias

- [1] Facultat d'Informàtica de Barcelona. <https://www.fib.upc.edu/>.
- [2] UPC - Universitat Politècnica de Catalunya. <https://www.upc.edu/es/>.
- [3] Pangea, Internet ètic i solidari. <https://pangea.org/>.
- [4] Electronic Reuse | Open Source Technology for reusing Digital Devices ensuring final recycling. <https://www.ereuse.org/>.
- [5] Github. eReuse/workbench: The eReuse.org Workbench is a toolset to assist with the diagnostic, benchmarking, inventory and installation of computers. <https://github.com/eReuse/workbench>.
- [6] Ellen Macarthur Foundation. Circularity indicators. An approach to Measuring Circularity. <http://www.ellenmacarthurfoundation.org/circular-economy/metrics>.
- [7] Circular economy. https://en.wikipedia.org/wiki/Circular_economy.
- [8] David A. Wheeler. Open Source Software / Free Software (OSS/FS or FLOSS) References. https://dwheeler.com/oss_fs_refs.html.
- [9] Andrew M. St. Laurent. *Understanding Open Source and Free Software Licensing*. O'Reilly Media, 2008.
- [10] Wikipedia: GNU General Public License. https://en.wikipedia.org/wiki/GNU_General_Public_License.
- [11] Wikipedia: The Open Source Definition . https://en.wikipedia.org/wiki/The_Open_Source_Definition.
- [12] Sheen S. Levine and Michael Prietula. Open Collaboration for Innovation: Principles and Performance. <https://ssrn.com/abstract=1096442>.
- [13] D. Franquesa, L. Navarro, D. López, X. Bustamante, and S. Lamora. Breaking Barriers on Reuse of Digital Devices. <http://hdl.handle.net/2117/83783>.
- [14] DeviceTag.io. <https://www.devicetag.io/>.
- [15] Github. eReuse/devicehub-teal. <https://github.com/eReuse/devicehub-teal>.
- [16] Github. eReuse/teal. <https://github.com/ereuse/teal>.
- [17] Github. eReuse/workbench-live. <https://github.com/eReuse/workbench-live>.
- [18] Debian. Live build Manpage. <https://manpages.debian.org/testing/live-build/live-build.7.en.html>.
- [19] Cambridge Dictionary. Scalability. <https://dictionary.cambridge.org/dictionary/english/scalability>.

- [20] Technology readiness levels. https://en.wikipedia.org/wiki/Technology_readiness_level#European_Union.
- [21] Lyonel Vincent. Hardware Lister (lshw). <http://lshw.ezix.org/>.
- [22] Debian. Details of package hwinfo. <https://packages.debian.org/stretch/hwinfo>.
- [23] Github. lshw: HardWare LiSter for Linux. <https://github.com/lyonel/lshw>.
- [24] Gitub. Hardware information tool code. <https://github.com/openSUSE/hwinfo>.
- [25] Debian. Details of package sysbench. <https://packages.debian.org/stretch/sysbench>.
- [26] Github. Repository of sysbench code. <https://github.com/akopytov/sysbench>.
- [27] Wikipedia: dd (Unix). [https://en.wikipedia.org/wiki/Dd_\(Unix\)](https://en.wikipedia.org/wiki/Dd_(Unix)).
- [28] GNU. Coreutils - GNU core utilities. <https://www.gnu.org/software/coreutils/>.
- [29] Github. Repository of phoronix test suite code. <https://github.com/phoronix-test-suite/phoronix-test-suite/>.
- [30] Wikipedia. MIT License. https://en.wikipedia.org/wiki/MIT_License.
- [31] Debian. Details of package stress. <https://packages.debian.org/stretch/stress>.
- [32] Smartmontool. Smartmontool official website. <https://www.smartmontools.org/>.
- [33] Wikipedia. shred (Unix). [https://en.wikipedia.org/wiki/Shred_\(Unix\)](https://en.wikipedia.org/wiki/Shred_(Unix)).
- [34] Archlinux. badblocks. <https://wiki.archlinux.org/index.php/Badblocks>.
- [35] Wikipedia. e2fsprogs. <https://en.wikipedia.org/wiki/E2fsprogs>.
- [36] Taiga.io. <https://taiga.io/>.
- [37] Github. Organization eReuse.org. <https://github.com/ereuse/>.
- [38] JetBrains. PyCharm. <https://www.jetbrains.com/pycharm/>.
- [39] GNU. Xorriso. <https://www.gnu.org/software/xorriso/>.
- [40] Free Software Foundation. Website. <https://www.fsf.org/>.
- [41] Github. Repository of squashfs-tool. <https://github.com/plougher/squashfs-tools>.
- [42] Libvirt. Website. <https://libvirt.org/>.
- [43] QEMU. Website. <https://www.qemu.org>.
- [44] Github. Prototipo código solución C. <https://github.com/eReuse/workbench-live/blob/devel/resources/modify-wb-iso.sh>.

[45] 2005 World Summit. https://en.wikipedia.org/wiki/2005_World_Summit.