



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Búsqueda y documentación de aplicaciones para el  
empoderamiento de la ciudadanía

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Vidal Cárcel, Juan Miguel

Tutor/a: Muñoz Escóí, Francisco Daniel

CURSO ACADÉMICO: 2021/2022



# Resumen

Existe un problema de independencia de algunas herramientas digitales dentro de las comunidades ciudadanas que tiene como raíz el no tener acceso de manera sencilla a herramientas alternativas libres que permitan su autogestión.

Este proyecto tiene como objetivo brindar de una manera técnica pero sencilla de un conjunto de herramientas libres que les ayuden a reducir o eliminar esa dependencia.

La iniciativa desarrollará la recopilación, el análisis y la documentación de herramientas cívicas libres que faciliten el empoderamiento ciudadano y solucionen problemas de gobierno abierto.

Estas herramientas se centrarán en ayudar a grupos de ciudadanos en temas de colaboración, transparencia digital y rendición de cuentas.

# Tabla de contenidos

1. Introducción	5
1.1 ¿Qué es el Civic Tech Test Lab?	5
1.2 Motivación	5
1.3 Objetivos del proyecto	6
1.4 Impacto esperado	6
1.5 Metodología	7
1.6 Estructura de este documento	11
2. Contexto tecnológico y social	13
2.1 Qué es la soberanía tecnológica	13
2.2 Qué es el software libre	14
2.3 Por qué es importante la soberanía tecnológica	14
2.4 Principales retos de la soberanía tecnológica	15
2.5 Proyectos relacionados (en progreso)	15
3. Análisis del problema	18
3.1 Riesgos de planificación	18
3.2 Análisis del marco legal y ético	18
3.3 Análisis de las licencias de las aplicaciones candidatas	19
3.4 Riesgos de recursos económicos	21
4. Diseño de la solución	22
4.1 Flujo de trabajo	22
4.2 Arquitectura del despliegue	24
4.3 Pruebas de funcionamiento	24

4.4 Análisis básico de seguridad de las aplicaciones analizadas	25
4.5 Estructura de la documentación	25
4.6 Tecnologías utilizadas	26
5. Documentación de las aplicaciones incluidas	28
5.1 Tablas de las aplicaciones documentadas	28
5.2 OpenRefine	30
5.3 RocketChat	33
5.4 FixMyStreet	38
5.5 Kanboard	41
5.6 LimeSurvey	44
5.7 RAWGraphs	48
5.8 Demokratian	50
6. Conclusiones	53
7. Bibliografía	55
8. Anexos	57
8.1 Motivación personal	57
8.2. Relación del trabajo con los Objetivos de Desarrollo Sostenible	57

# 1. Introducción

## 1.1 ¿Qué es el Civic Tech Test Lab?

El Civic Tech Test Lab (CTTL) es un proyecto realizado por la Cátedra de Tecnología Cívica y Empoderamiento (TCE) promovida por la Delegació de Transparència i Govern Obert de la Diputació de València junto a la Universitat Politècnica de València. La iniciativa tiene como objetivo la recopilación, el análisis y la documentación de herramientas cívicas que faciliten el empoderamiento ciudadano y solucionen problemas de gobierno abierto [16].

La TCE se encarga de desarrollar diferentes ideas bajo los mismos principios de empoderamiento y libertad, contribuyendo a que la información y la tecnología se acerque a los usuarios que la necesiten y esté a disposición de la ciudadanía.

En esta beca pasé 6 meses buscando diferentes aplicaciones que pudieran ayudar a grupos de ciudadanos a resolver problemas de coordinación y colaboración mediante herramientas tecnológicas libres. Esta idea nació con el objetivo de empoderar al ciudadano facilitándole el camino hacia herramientas que pudieran resultarle útiles y bajo esta filosofía solo se cribaron las aplicaciones que estuvieran bajo una licencia de código libre. Durante este tiempo compaginé el CTTL con mis estudios habituales del Grado de Ingeniería Informática.

## 1.2 Motivación

La ciudadanía cada vez se apoya más en herramientas digitales para ayudarse en procesos importantes del día a día. Con el avance de la tecnología y la continua innovación en aplicaciones digitales se confía más en aplicaciones terceras privadas las cuales se escapan a nuestro control y sacrificamos libertad personal en pos de delegar la responsabilidad que tenemos respecto a conocer las tecnologías que tanto influyen en nuestras vidas.

Muchas comunidades de ciudadanos quieren recuperar su soberanía tecnológica y poder tener una autogestión digital de sus herramientas y datos pero encuentran difícil sustituir y hacer suyas alternativas y herramientas libres.

Con este proyecto se intenta demostrar que existen herramientas libres de empoderamiento ciudadano y acercar su uso facilitando documentación y vigencia.

## 1.3 Objetivos del proyecto

El objetivo del proyecto fue el de crear un repositorio de herramientas libres de organización y coordinación centradas en el empoderamiento ciudadano para mejorar la soberanía tecnológica de los ciudadanos y dar visibilidad a alternativas libres.

Para cumplir este objetivo se buscó, analizó y documentó herramientas para grupos de ciudadanos cuyo algún integrante tuviera conocimientos muy básicos de servicios en la nube o de gestión de servidores.

Para ayudar al cumplimiento y seguimiento de la meta general del proyecto se propuso que las tres etapas principales del proyecto tuvieran tres objetivos que fueran cumpliéndose en orden.

- Etapa 1: En la primera etapa dedicamos el tiempo a decidir qué tipo de herramientas sería adecuado que buscásemos y qué tipo de licencias estarían alineadas con la filosofía de la cátedra. También investigamos qué servicios cloud podrían ser más adecuados para este proyecto principalmente por curva de aprendizaje aunque también por precio. El objetivo de esta etapa fue sacar una lista de pautas para cribar las aplicaciones que fuéramos encontrando.
- Etapa 2: En la segunda etapa dedicamos el tiempo a la búsqueda de las aplicaciones que reuniesen los criterios decididos en la etapa uno y a la documentación de su despliegue. El objetivo de esta etapa era tener una lista de aplicaciones documentadas que hubiéramos probado.
- Etapa 3: El objetivo de la tercera etapa fue el de hacer accesible el repositorio del proyecto y darlo a conocer.

A partir de la etapa 3 tendremos ya la documentación desarrollada a lo largo del proyecto disponible para toda la comunidad que cumplirá con el objetivo de reunir un grupo de herramientas fiables y libres que ayuden a reducir la dependencia de herramientas privativas de autogestión.

El proyecto fue llevado de inicio a fin en los seis meses que duró la beca así que no hubo problemas de integraciones o de cierre del proyecto.

## 1.4 Impacto Esperado

Se espera que esta iniciativa ayude a diferentes comunidades de ciudadanos a organizarse y plantear soluciones tecnológicas a problemas comunes de colaboración, coordinación y comunicación. Un posible ejemplo de algunas de estas comunidades son los grupos de vecinos que necesitan soluciones tecnológicas para organizarse y atacar diferentes problemas de una finca o un barrio. El CTTL espera acercar estas herramientas al ciudadano que pueda necesitarlo de manera sencilla y clara.

Otro impacto secundario más abstracto esperado es también el de potenciar y dar visibilidad a este tipo de proyectos. Estas iniciativas ayudan a reducir la desigualdad tecnológica aumentando el conocimiento accesible y libre.

## 1.5 Metodología

La metodología del proyecto varió entre las etapas uno y dos. En la primera etapa el método de trabajo consistió en diversas reuniones donde aportábamos diferentes perspectivas de los problemas que pueden tener grupos de ciudadanos y qué soluciones tecnológicas podían ayudar a solventarlos. Una vez decidido esto se acordó qué características debían tener las licencias de estas herramientas, se decidió la estructura básica común de la documentación que se desarrollaría para cada aplicación y por último se decidió qué servicio en la nube convenía para el desarrollo del proyecto.

En este proyecto trabajábamos dos personas y se tomaron las decisiones expuestas en el párrafo anterior junto al supervisor del proyecto Diego Álvarez Sánchez, profesor de la UPV y colaborador del CTTL. Cuando tuvimos un acuerdo inicial de cómo desarrollaríamos el proyecto, se trabajó por separado. Se mantuvieron reuniones semanales para actualizarnos y comentar dificultades, dudas o bloqueos que estuvieran apareciendo a lo largo del proyecto. En este documento solo se incluyen las aplicaciones desplegadas y desarrolladas por mí. En el repositorio del CTTL se podrán encontrar todas las herramientas que se documentaron para el proyecto.

Se calculó que aproximadamente se utilizaría el primer mes para asentar las bases del proyecto y aprender la metodología y tecnologías que utilizaríamos y se utilizarían los siguientes tres o cuatro meses para el descubrimiento, despliegue y documentación de las aplicaciones dejando una o dos semanas para finalmente hacer una presentación del proyecto. En los siguientes diagramas de Gantt por meses se pueden ver las distintas fases del proyecto ya actualizadas con las aplicaciones que se escogieron. Se puede observar que algunas aplicaciones costaron más de desplegar y analizar por su complejidad. En los diagramas se incluye el tiempo empleado descartando otras aplicaciones que no cumplieran los requisitos del proyecto en las tareas del tipo “Descubrimiento de la aplicación ...” de la segunda etapa. Se muestra también en amarillo los días que tuvimos reuniones de sincronización y coordinación.

La segunda etapa sigue una pauta que se repita donde cada aplicación tiene un ciclo de tres fases: descubrimiento, despliegue y documentación.



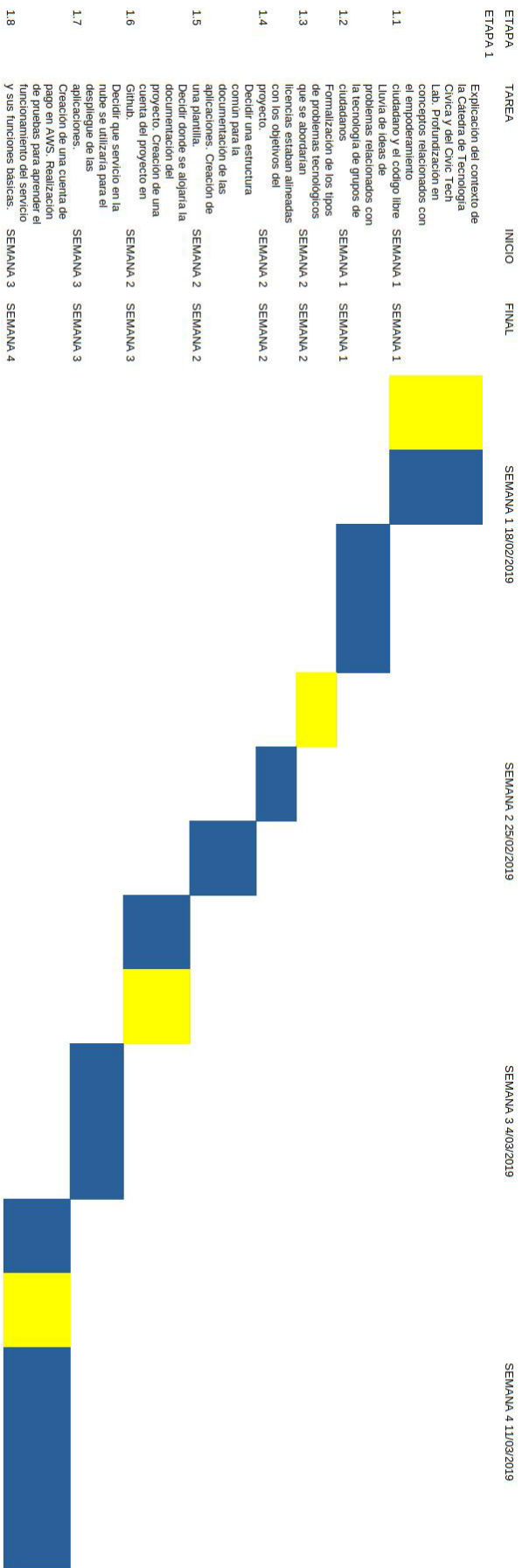


Diagrama 1. División temporal de tareas del primer mes

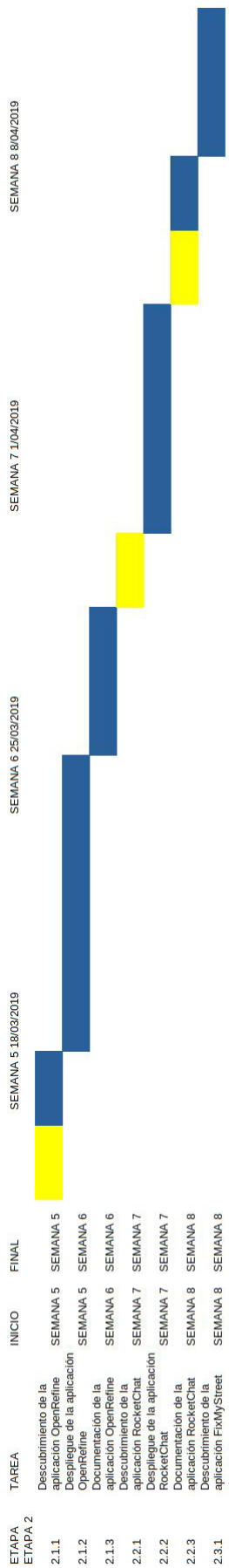


Diagrama 2. División temporal de tareas del segundo mes

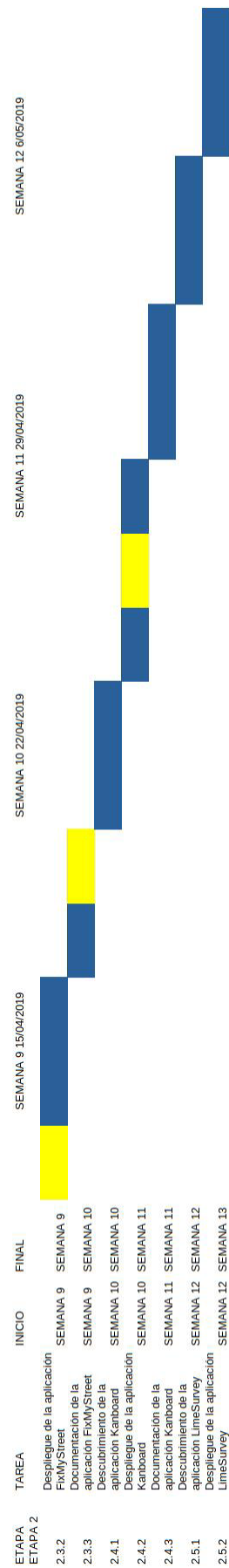


Diagrama 3. División temporal de tareas del tercer mes

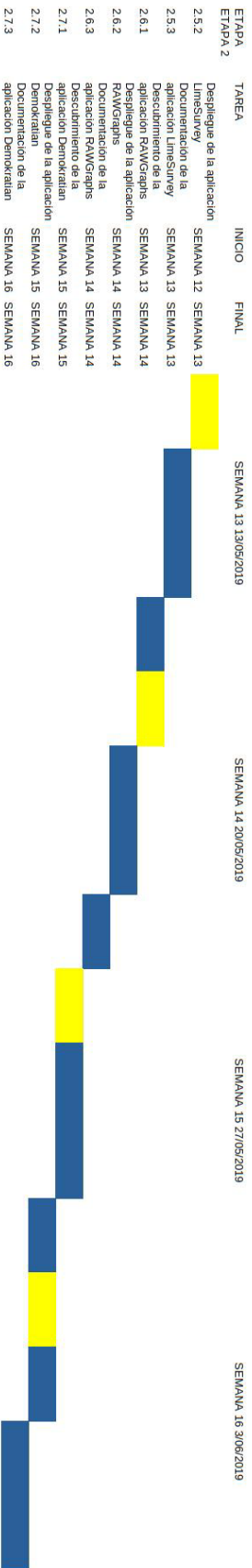


Diagrama 4. División temporal de tareas del cuarto mes



Diagrama 5. División temporal de tareas del quinto mes

Para el despliegue y documentación de la segunda etapa seguimos los siguientes tres pasos por cada herramienta.

- **El primer paso fue el de descubrimiento.** Se buscaron herramientas que resolvieran problemas que se hubieran comentado en la etapa anterior y si estas herramientas eran privativas se priorizaban alternativas libres. En este paso se filtraron aplicaciones poco mantenidas o que requiriesen de un nivel de conocimientos demasiado alto para el usuario técnico medio.

- **El segundo paso fue el de despliegue.** En este paso se creaba una máquina virtual en el servicio en la nube y se desplegaba la aplicación elegida de manera que fuera completamente funcional y lista para su uso. Si durante el proceso de despliegue se detectaban complicaciones relacionadas con problemas sobre el desarrollo de la herramienta esta se descartaba. Como tenía algo de experiencia previa con los servicios de Amazon Web Services (AWS) casi todos los proyectos fueron desplegados en instancias EC2 de Amazon.

- **El tercer paso fue el de documentación.** En este paso, si la herramienta había pasado los filtros anteriores, se documentaba tanto en castellano como en valenciano y se añadía al repositorio del CTTL.

## 1.6 Estructura de este documento

El documento se distribuye en las siguientes secciones:

- **Introducción.** Esta sección proporciona una introducción al proyecto en relación a la motivación, el impacto esperado, los objetivos y metodología empleada.

- **Contexto tecnológico y social.** En esta sección se tratan conceptos claves para contextualizar el proyecto en el contexto actual. También se expondrán las ventajas que aporta este proyecto en contraste con otras soluciones.

- **Diseño de la solución.** El objetivo de esta sección es explicar como se van a llevar a cabo las diferentes partes del proyecto, el flujo de trabajo y las diferentes tecnologías que se utilizaron.

- **Documentación de las aplicaciones incluidas.** En este apartado se expone la documentación de las aplicaciones que fueron desplegadas satisfactoriamente y consideradas aptas para el desarrollo de los objetivos del proyecto.

- **Conclusiones.** Aquí aparecen las reflexiones relativas al desarrollo del proyecto y al cumplimiento de los objetivos de este.

- **Bibliografía.** En esta sección aparecen todas las fuentes usadas en el proyecto.

- **Anexos.** En los anexos se han añadido datos de interés personal pero que no son indispensables para comprender el proyecto

## 2. Contexto tecnológico y social

En este capítulo se van a exponer algunos conceptos clave para entender la situación social que hacen necesarias soluciones como las propuestas en este proyecto. Se introducirán de manera breve las corrientes de soberanía tecnológica y de código libre junto con sus principales motivaciones y retos.

También se detallarán las principales ventajas que aporta este proyecto en comparación con otras soluciones actuales.

### 2.1 Qué es la soberanía tecnológica

La soberanía tecnológica es una corriente de pensamiento o perspectiva política que postula la necesidad de un empoderamiento por parte de la ciudadanía en el entorno de las nuevas tecnologías.

Los objetivos principales de esta corriente son:

- Conseguir que la ciudadanía sea consciente y responsable del uso de estas tecnologías y de su consumo.

- Desarrollar y difundir alternativas autogestionadas y libres especialmente en los ámbitos más comunes para los usuarios no necesariamente técnicos. Ejemplos de estos ámbitos pueden ser: comunicaciones, sistemas operativos, herramientas de ofimática, servicios relacionados con internet como navegadores, redes sociales... Entre otros.

Este movimiento está muy relacionado y apoyado por otros tales como el software libre o la gobernanza individual de los datos, el decrecimiento...

También se suele unir este concepto a distintas corrientes de pensamiento relacionadas con el decrecimiento. Estas fomentan un uso más responsable de la tecnología y condenan el “desarrollo por el desarrollo” buscando que la tecnología crezca de manera responsable y se use para solucionar problemas sociales y ecológicos en lugar de, muchas veces, expandirlos.

## 2.2 Qué es el software libre

El control del usuario sobre un programa se basa en cuatro libertades fundamentales [7].

- La libertad de ejecutar el programa con el propósito que se desee.
- La libertad de acceder, estudiar y cambiar el código fuente del programa para que se ejecute de la forma que se desee.
- La libertad de hacer y distribuir copias del programa original de la forma que se desee.
- La libertad de hacer y distribuir copias de tus versiones modificadas en el momento que se desee.

De esta forma si algún programa no cumpliera una o varias de estas libertades se consideraría un programa privativo.

Algunas ventajas del software libre para usuarios potenciales son:

- Es gratuito y existen formas de donaciones y contribuciones para mantener estos proyectos.
- Uso de estándares. Esto hace más sencillo la adaptación e interoperatividad de las aplicaciones libres.
- Fuerte soporte general proporcionado por una amplia comunidad.
- Mayor seguridad y privacidad por que se tiene el control total de lo que se ejecuta en tu ordenador o servidor.

Si bien es cierto que como desventajas se pueden encontrar la necesidad de ciertos conocimientos técnicos básicos o la “garantía” de las aplicaciones estos dos problemas son parte de los objetivos de este proyecto [4]. Mediante este proyecto se quiere facilitar esta información técnica de manera sencilla y filtrar las aplicaciones más útiles y mantenidas por lo que estas desventajas no afectan al propósito del proyecto.

## 2.3 Por qué es importante la soberanía tecnológica

La sociedad está cada vez más digitalizada y con la aceptación de tecnologías facilitadoras como móviles y ordenadores en el día a día se ha delegado y confiado en muchas aplicaciones partes importantes de nuestras vidas. El problema surge en que en la mayoría de los casos no se conoce qué hacen estas aplicaciones, solo conocemos su “funcionalidad” más directa y eso reduce la libertad del usuario.

“La aplicaciones privativas son controladas por gobiernos y empresas, las aplicaciones libres las controlan sus usuarios” Richard Stallman

Esto significa que si el usuario no controla el programa de cierta manera el programa controla al usuario. Cuando se ceden operaciones o necesidades a una aplicación privativa (en local o en un servidor con metodología SaaS) se confía de manera ciega que hará exactamente lo que se le pida y nada más y ello conlleva una pérdida de libertad [8].

Se puede explicar esta cesión “voluntaria” de libertad por el rápido crecimiento y cambio de la tecnología pero también es cierto que la mayoría de veces la poca interoperabilidad y malas prácticas de empresas de desarrollo de software privativo evitan que se extiendan otras alternativas libres.

La soberanía tecnológica es fundamental ya que cuanto más necesitemos de tecnologías digitales para procesos importantes de nuestras vidas más importante es que tengamos un control real sobre ellas. También es importante comentar que todas las personas son expertas en su propia relación con la tecnología, por ello cualquiera puede contribuir en analizarlas con el objetivo de reinventarlas y adoptarlas [3].

## **2.4 Principales retos de la soberanía tecnológica**

La soberanía tecnológica es complicada de alcanzar ya que tiende a combinar retos tanto políticos como tecnológicos [1].

Algunos de estos retos son:

- Movimientos de monopolio digital de empresas de aplicaciones privativas.
- Exclusión de alternativas digitales libres en dispositivos hardware privativos.
- Poca eficiencia en repositorios libres y dificultades lingüísticas.
- Poca documentación y excesiva informalidad en la estructura de algunos proyectos libres.
- Carencia de motivación para desarrollar y mantener herramientas libres.

## **2.5 Proyectos relacionados**

En esta sección se expondrán diferentes proyectos relacionados con los objetivos de este documento y sus diferencias con este.

En relación con otros trabajos de fin de grado podemos encontrar una investigación muy interesante desarrollada por Omayra Vidal Montblanch y hecho en la Universidad Politécnica de Valencia titulada “Empoderamiento ciudadano en Benimaclet: autogestión y movimientos sociales” [9].

En este trabajo Omayra hace un estudio de las necesidades de cinco grupos diferentes del ciudadanos del barrio valenciano de Benimaclet y las dificultades que se encuentran



para la coordinación, la autogestión y el espacio que necesitan para el desarrollo de sus actividades. Este estudio revela que existen diferentes retos para estas organizaciones y propone solucionar algunas de las relacionadas con dificultades técnicas mediante herramientas de reorganización participativas y la incorporación de canales de autoformación para conseguir mayor soberanía e independencia tecnológica.

Es un trabajo muy interesante que ataca problemas comunes a los expuestos en esta memoria. En el estudio de Omayra se tienen en cuenta situaciones y retos más amplios que los descritos en este proyecto por ello creo que lo expuesto en las siguientes páginas podría ser de utilidad para complementar estudios como este teniendo también en cuenta que los problemas y soluciones expuestos en este desarrollo pueden aplicarse a diferentes comunidades no limitadas a Benimaclet o a las actividades sociales de los grupos que trata en su memoria.

También se encontraron muchas páginas y aplicaciones dedicadas a exponer alternativas libres pero se llegó a la conclusión que no cumplían los objetivos que se buscaban en este proyecto por diferentes motivos. Las principales fueron:

- **Opensourcealternative.to.** Esta página ofrece un gran número de alternativas libres para muchas necesidades. Tiene un sistema de puntuación y ofrece un sistema de filtros. Es una página ampliamente utilizada. Se enfoca principalmente a herramientas y usos técnicos como alternativas libres a bases de datos, APIs, sistemas de monitorización... No se enfoca en aplicaciones con una funcionalidad y un propósito definido si no que lo junta con paquetes, librerías y entornos para el desarrollo sin ofrecer descripciones para estos por lo que se necesita un conocimiento previo sobre estas aplicaciones. Además, uno de los objetivos de este proyecto es su accesibilidad y Opensourcealternative existe únicamente en inglés [11].

- **Opensourcesoftwaredirectory.com.** Esta página también ofrece muchas alternativas libres para diferentes proyectos. Además divide las aplicaciones en cuatro grupos: software para uso no técnico (Home users), software para empresas (Business), para el desarrollo (Developers) y para administración (Admins). Esta clasificación ayuda a clasificar fácilmente las alternativas para usuarios no técnicos facilitando su búsqueda como es el objetivo de este proyecto. Además solo lista aplicaciones libres y no "gratis". Por otro lado no se centra en ninguna funcionalidad o grupo de usuarios particular, las descripciones de las aplicaciones son cortas y no están avaladas con despliegues o pruebas propios y solo se encuentra en inglés. Es una página de referencia de uso muy extendido [15].

- **Opensource.com.** Esta página fue lanzada por el CEO de la empresa Red Hat Jim Whitehurst en 2010 y actualmente es mantenida por esta empresa. Es una página interesante porque aunque está centrada en herramientas libres para desarrolladores tiene una sección llamada "Proyectos y aplicaciones" que incluye diferentes alternativas libres de diversas temáticas. Su lista es un poco confusa ya que incluye desde alternativas para gestión del tiempo, gestión de correos, presentaciones hasta alternativas a videojuegos como "Minecraft". Aunque en las listas incluyen pequeñas descripciones de las aplicaciones que exponen no incluyen ni despliegues ni pruebas ni ningún tipo de filtro para catalogarlas. La hemos incluido en esta sección para mostrar que

existen páginas centradas en el software libre y de proyectos muy reconocidos que se quedan muy cortas en el ámbito de acercar soluciones útiles para el empoderamiento ciudadano y se limitan a ofrecer una lista y descripciones a veces demasiado técnica de herramientas sin criterios definidos [14].

### 3. Análisis del problema

Se observó que a ciertas comunidades les faltaban herramientas tecnológicas en su día a día o estaban forzadas a utilizar aplicaciones privativas para su gestión.

El problema se encontró en que muchas comunidades ciudadanas aunque quieran utilizar herramientas libres para su gestión y desarrollo muchas veces este conocimiento es de difícil acceso y puede requerir de muchas horas de investigación previa por lo que tendían a utilizar herramientas privadas. El CTTL se crea con la finalidad de actuar como facilitador en este paso, de listar una serie de herramientas útiles y probadas que ahorrarían mucho tiempo en esa etapa.

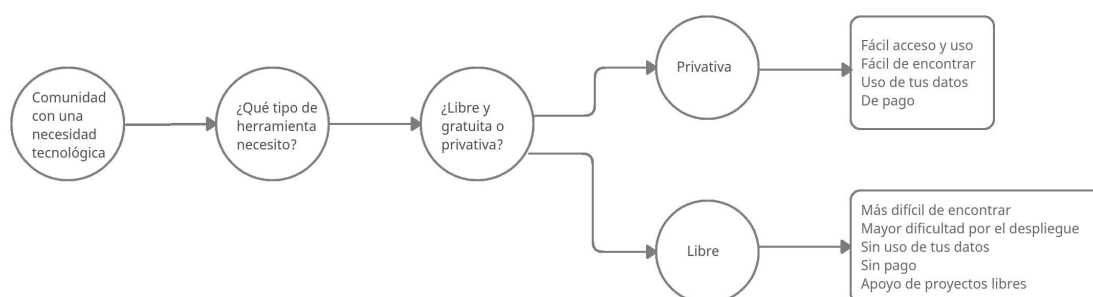


Diagrama 6. Flujo de análisis del problema

#### 3.1 Riesgos de planificación

El proyecto tenía como limitación el tiempo que se podía emplear para desarrollarlo. El proyecto estaba pensado para durar 6 meses que era la duración del contrato de prácticas.

Dada la cantidad de aplicaciones de código libre que podríamos documentar relacionadas con el tema nos propusimos documentar entre 8 y 12 aplicaciones que cumplieran con nuestros criterios para tener suficiente cobertura y tiempo para la etapa 3.

#### 3.2 Análisis del marco legal y ético

Se decidió que todas las aplicaciones elegibles para el CTTL tendrían que ser bajo licencias “libres” para facilitar y promover el uso de herramientas que apoyen la autogestión e independencia digital [2].

Los propios recursos del CTTL se distribuyen bajo la licencia GNU Free Documentation License que se consideró la mejor licencia para este proyecto por los siguientes motivos:

- **Gratis no significa libre:** las licencias libres garantizan que no solo sea gratis si no que abogan por una libertad más amplia.

- **Total libertad:** esta licencia garantiza que el proyecto CTTL puede usarse bajo cualquier contexto y sin ninguna restricción en cuanto a su uso, modificación o comercialización.

- **Apoyo al software libre:** cualquier proyecto que quiera utilizar o expandir alguna parte o la totalidad del CTTL tendrá que publicarse bajo la misma licencia GNU.

- **Responsabilidad:** la licencia garantiza que el CTTL no tiene responsabilidad sobre el mal uso o errores propios del proyecto.

Esta licencia está ampliamente adoptada y es de las más usadas en referente a documentación libre.

### 3.3 Análisis de las licencias de las aplicaciones candidatas

La soberanía tecnológica tiene muchos puntos en común con la filosofía del código libre por ello se tuvo que hacer una criba de las aplicaciones también por la permisividad de su licencia. Se admitieron una amplia variedad de aplicaciones con licencias diferentes siempre teniendo en cuenta que estas estuvieran alineadas con la filosofía del código libre o abierto [12 y 13].

Existen diferentes corrientes relacionadas con el código libre y qué licencias realmente pueden considerarse parte de este. La discusión sobre qué es “libre” es muchas veces motivo de conflicto dentro de la comunidad y tiene sobretodo dos grandes vertientes en cuanto a licencias: los proyectos con licencias permisivas y los proyectos con licencias copyleft.

Las licencias permisivas y de copyleft tienen algunas diferencias. La diferencia principal es que la licencias permisivas no obligan a que el código modificado sea publicado bajo la misma licencia que el original mientras que las licencias copyleft sí que obligan a que el código modificado herede la licencia del original.

El debate sobre cuáles son realmente libres o no puede ser largo y tiene muy buenos argumentos a favor y en contra. Desde la perspectiva del proyecto se decidió que incluiríamos ambos tipos ya que ambos aportaban valor a la soberanía tecnológica de los ciudadanos.

Las licencias de las aplicaciones que finalmente se añadieron a la documentación fueron las siguientes:

#### **BSD-3-Clause - Permissive**

Esta licencia permite la distribución del código original o modificado de la forma que se desee siempre que este código distribuido retenga el copyright de esta licencia y no se usen los nombres de los autores originales sin su permiso [12].

## **Apache 2.0 - Permissive**

Esta licencia te permite la distribución del código original o modificado de la forma que se desee siempre que este incluya

- El aviso original de copyright
- Una copia de la licencia
- Una lista de cambios hechos al código original
- Una copia del archivo NOTICE original si lo tuviera

El código modificado no es necesario que se distribuya bajo la licencia de Apache 2.0

## **General Public License (GPL) 2.0 - Copyleft**

Esta licencia te permite la distribución del código original o modificado siempre que mantengas una lista de cambios de los archivos originales y que el producto resultante, junto a su documentación, sea distribuido también bajo la licencia GPL 2.0.

## **General Public License (GPL) 3.0 - Copyleft**

La licencia GPL 3.0 tiene algunas diferencias respecto a su versión 2.0. Su mayor diferencia es que deja claro explícitamente que los derechos de patente de las distribuciones modificadas del código original no dependen del proyecto original.

Esto implica que mientras que con la licencia GPL 2.0 el problema los derechos de patente de las modificaciones sobre el código original estaba tratado de manera general y abstracta la GPL 3.0 deja claro que el proyecto original no tiene derechos sobre el uso, venta, distribución... de las modificaciones de este.

## **Affero General Public License (AGPL) 3.0 - Copyleft**

La licencia AGPL nace de la licencia GPL y mantiene los mismos derechos y obligaciones extendiéndola con una cláusula que obliga a publicar el código si este se ejecuta en una red de servidores.

## **MIT License – Permissive**

La licencia MIT permite el uso, modificación y distribución del código original como se desee siempre que se incluya una copia de la licencia y el aviso de copyright original. El autor del código original tampoco se puede hacer responsable legalmente sobre él.

### **3.4 Riesgos de recursos económicos**

No había una limitación concreta sobre el gasto de las máquinas virtuales ya que en la mayoría de aplicaciones se utilizaron generalmente máquinas virtuales t2.micro de AWS cuyo consumo es mínimo y se eliminaban poco después de su despliegue. Igualmente existían diversas alertas que hubieran impedido que su gasto superara los 50€ por mes. Cualquier anomalía se hubiera comentado en las reuniones semanales.

# 4. Diseño de la solución

Durante las dos primeras semanas de la beca se discutieron diferentes aproximaciones para desarrollar el proyecto. Durante este tiempo formalizamos el flujo de trabajo para los próximos meses, la estandarización junto con las características del despliegue y la estructura de la documentación.

## 4.1 Flujo de trabajo

El flujo de trabajo general constó de tres fases en las cuales las aplicaciones propuestas podían ser rechazadas por diferentes motivos.

En la fase de descubrimiento las aplicaciones podían ser rechazadas por:

- Por motivos éticos si su licencia no era totalmente libre o incluían software poco ético de recopilación de datos.
- No aportaban lo suficiente al proyecto porque otra aplicación cumplía su propósito mejor o su utilidad no se alineaba con los objetivos del proyecto.
- No tenían mantenimiento y estaban deprecadas.

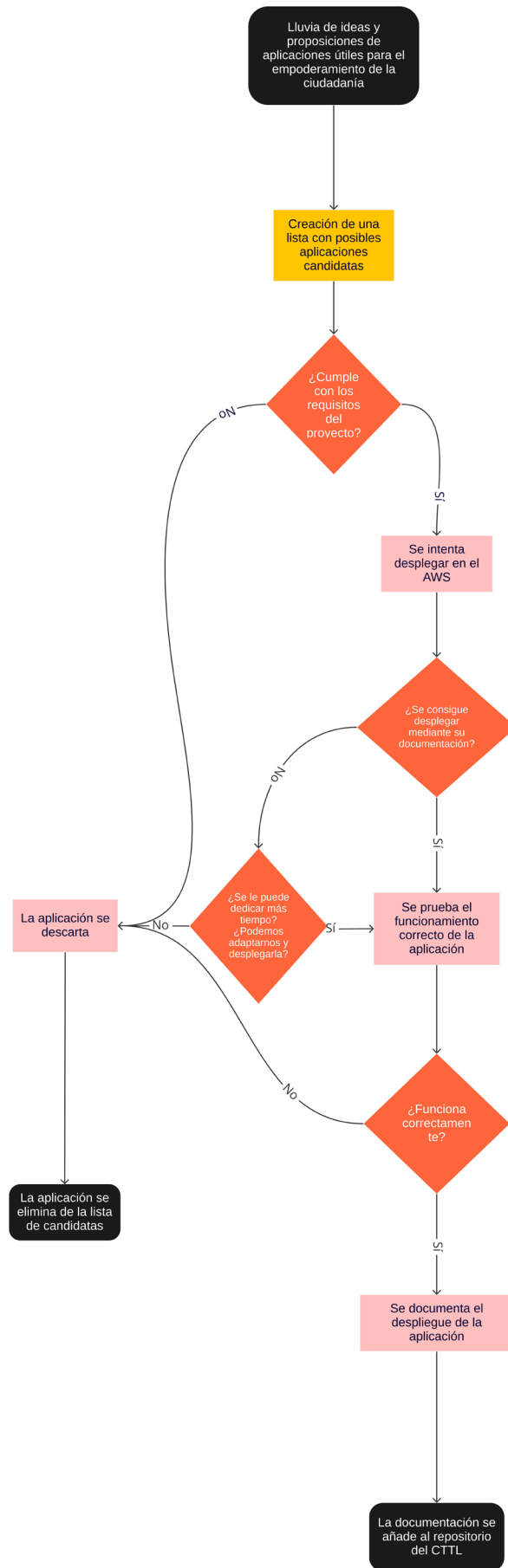
En la fase de despliegue las aplicaciones podían ser rechazadas por motivos técnicos o de tiempo:

- Si la documentación de la aplicación era poco clara o incorrecta y el coste de tiempo en adaptarla no era asumible.
- Si al probarla no funcionaba correctamente y no se debiera a un error de la configuración del despliegue.

En la fase de documentación no se rechazó ninguna aplicación porque todas las aplicaciones que llegaban a esta fase ya habían sido validadas en las fases anteriores.

En fases anteriores sí que llegaron a rechazarse diversas aplicaciones aunque siempre estas tenían una utilidad limitada. Las aplicaciones más completas y útiles normalmente están bien mantenidas y tienen una documentación más o menos buena ya que suelen tener una comunidad que ayuda a que estén actualizadas.

En cualquier caso si alguna aplicación interesante para el proyecto tuvo mala documentación se intentó desplegar y anotar las mejoras a la documentación oficial que se aplicaron para desplegarla. Este fue el caso de una aplicación que se expone más adelante llamada Kanboard cuya documentación estaba un poco desactualizada.





## 4.2 Arquitectura del despliegue

Para poder validar las aplicaciones que estuvieran alineadas con los objetivos del proyecto se intentó estandarizar el proceso del despliegue, la arquitectura cloud necesaria y su posterior documentación. En el desarrollo del proyecto se encontraron situaciones en las que tuvimos que ser flexibles y adaptar este proceso pero en general se mantuvo según las siguientes especificaciones.

- Instancias virtuales t2.micro de AWS.
- Sistema operativo Ubuntu 18.04
- Última versión de docker si la aplicación lo permitía.

Se decidió utilizar el servicio cloud de Amazon Web Services por ser junto a Google Cloud el más sencillo de aprender y, en el momento del desarrollo del proyecto, también el más barato.

Se decidieron probar estas aplicaciones en instancias pequeñas para comprobar que pudieran ser desplegadas también en hardware poco potente como ordenadores antiguos o Raspberry Pis.

Se decidió que el sistema operativo que usaríamos para desplegar las aplicaciones sería Ubuntu por ser la distribución linux más accesible y conocida para la mayoría de personas. La versión 18.04 era la versión más estable de Ubuntu en el momento del desarrollo del proyecto.

Las dependencias de cada aplicación las instalábamos individualmente si el proyecto no contaba con una imagen docker.

La persistencia de las diferentes aplicaciones se intentó mantener dentro de la misma máquina virtual o contenedor pero se valoró si era necesario externalizarlo al servicio de almacenamiento S3 de AWS.

## 4.3 Pruebas de funcionamiento

Para probar el correcto funcionamiento de las aplicaciones desplegadas se probaron principalmente su persistencia y su capacidad para aguantar el uso simultáneo de diferentes usuarios sin percibir inconsistencias.

La persistencia se comprobó cambiando el estado de las aplicaciones y analizándolo de nuevo horas más tarde. En el caso de utilizar un servicio docker se paraba el contenedor y se volvía a iniciar para comprobar su estado, de no utilizarlo se intentaba matar el proceso y volver a levantar la aplicación.

El uso simultáneo de varios usuarios se emuló mediante diferentes conexiones simultáneas a la aplicación y comprobando manualmente que los cambios de diferentes usuarios no llevarsen a estados inconsistentes o de error.

## 4.4 Análisis básico de seguridad de las aplicaciones analizadas

Se decidió que un análisis extenso sobre la seguridad de las aplicaciones probablemente aumentaría mucho la complejidad del proyecto sin aportar a los objetivos del proyecto lo suficiente como para justificar el tiempo que se debía invertir para hacerlo en profundidad.

Por ello se acordó durante la primera etapa del proyecto que durante el despliegue y las pruebas de funcionamiento de las aplicaciones se comprobaran que estas tuvieran un mínimo de seguridad. El proceso podía variar un poco de una aplicación a otra pero de manera básica se comprobaron tres cosas.

- Que los servicios que proporcionaban las aplicaciones se sirvieran sobre https en lugar de http por defecto. Recomendar lo primero de no ser así.

- Que hubiera alguna comprobación sobre los datos que se enviaran al servidor en el caso de las aplicaciones con servicios frontend y backend para evitar que pudiera haber un fallo que permitiera acceder a los datos almacenados por el servicio.

- Que los paquetes, módulos y librerías que se utilizaran en la aplicación estuvieran actualizadas. Normalmente esto se revisó comprobando los requisitos de las aplicaciones durante la instalación y revisando brevemente los package.json si los tenía el servicio.

De cualquier forma se asumió que la seguridad no era bloqueante en cuanto al filtro de las aplicaciones candidatas siempre que estas estuvieran bien mantenidas. Un estudio más complejo se hubiera salido del alcance del proyecto.

## 4.5 Estructura de la documentación

En las dos primeras semanas previas a las fases expuestas se formalizó también los apartados que debía contener la documentación de cada aplicación. El objetivo de la documentación de cada aplicación era que fuera lo más breve y sencilla posible para facilitar su seguimiento. Los apartados que se desarrollaron finalmente fueron:

1. Tabla de contenidos o índice.
2. Descripción. Breve descripción de la aplicación documentada.
3. Autoría. Quienes lo desarrollaron y contexto si lo requiere.
4. Licencia.
5. Requerimientos Hardware. Requerimientos mínimos probados para su uso.

6. Instalación.
  - 6.1 Entorno.
  - 6.2 Dependencias.
  - 6.3 Despliegue.
7. Documentación oficial.
8. Comentarios adicionales. Únicamente de ser necesarios.
9. Seguridad. Análisis básico de la seguridad que pueda proporcionar la aplicación.
10. Valoración. Corta valoración personal de la aplicación y su despliegue.

Durante el desarrollo del proyecto las aplicaciones se fueron clasificando en tres grupos según su funcionalidad en relación al gobierno abierto. Los tres grupos son: Transparencia y datos, Colaboración y Rendición de cuentas.

## 4.6 Tecnologías utilizadas

En esta sección se expondrán diferentes tecnologías claves relacionadas con los proyectos documentados.

### **AWS**

Amazon Web Services (AWS) es un conjunto de servicios de computación en la nube ofrecidos por la empresa Amazon.

### **EC2**

Amazon Elastic Compute Cloud (EC2) es un servicio que ofrece AWS. Consiste en una instancia en un servidor virtual que actúa como una máquina virtual en la nube.

### **PHP**

Personal Home Page o Hypertext Preprocessor (PHP) es un lenguaje de programación de código abierto multiparadigma habitualmente utilizado para el desarrollo de aplicaciones web ejecutándose en el servidor.

### **Docker**

Docker es un programa que permite empaquetar aplicaciones junto con sus dependencias en unidades estándar que reciben el nombre de contenedores. Es muy útil para estandarizar procesos de despliegue porque hace fiable la ejecución de cualquier aplicación encapsulada en cualquier entorno con docker y reduce largos procesos de instalación y despliegue a unas pocas líneas de comandos.

## **Apache**

Es un programa de código abierto para la gestión de servidores HTTP. Es un proyecto ampliamente utilizado y mantenido por la comunidad. Es altamente configurable y flexible.

## **Open 311**

Es una tecnología de localización y comunicación de código libre para el desarrollo de aplicaciones de gobierno abierto. Entre sus varias funciones una de ellas es la de comunicación asíncrona de información útil para la comunidad. Entre los usos de la ciudadanía que le han dado a esta API está el de comunicar un accidente que provoca un atasco en una carretera, avisar de un bache que hay que reparar, fallos en alguna instalación pública... Es una herramienta muy útil pero está dirigida sobretodo a Estados Unidos, donde se desarrolló. Es una tecnología con mucho potencial si se despliega un servicio parecido en España. Sus actualizaciones son reducidas por el número de colaboradores pero sigue en mantenimiento [10].

# 5. Documentación de las aplicaciones incluidas

En este capítulo se presentarán las aplicaciones que fueron finalmente escogidas para su análisis, que pudieron desplegarse sin problemas y funcionaban correctamente.

## 5.1 Tablas de las aplicaciones documentadas

Estas tablas recogen la información resumida (Autor, licencia y temática) de los tres grupos de aplicaciones que fueron documentadas.

- Transparencia y datos

<b>Servicio</b>	<b>Autor</b>	<b>Licencia</b>	<b>Temática</b>
OpenRefine	OpenRefine	BSD-3-Clause	Filtrado de datos, tratamiento de datos
RAWGraphs	DensityDesign	Apache 2.0	Gráficos, tratamiento de datos
LimeSurvey	LimeSurvey	GPL 2.0	Encuestas, gráficos, tratamiento de datos

- Colaboración

<b>Servicio</b>	<b>Autor</b>	<b>Licencia</b>	<b>Temática</b>
Citizen OS	Citizen OS	Apache 2.0	Colaboración, votos, gobierno abierto
Decidim	Decidim	AGPL v3	Plataforma digital democrática, votos, gobierno abierto
Shareabouts	OpenPlans	GPL 3.0	Mapas colaborativos, open data
Fixmystreet	MySociety	AGPL v3	Mapas colaborativos, open data
RocketChat	RocketChat	MIT	Coordinación, mensajería
Kanboard	Kanboard	MIT	Colaboración, división de tareas, kanban
Demokratian	Demokratian	GPL 3.0	Colaboración, herramienta de voto

- Rendición de cuentas

<b>Servicio</b>	<b>Autor</b>	<b>Licencia</b>	<b>Temática</b>
Del dicho al hecho	Fundación Ciudadano Inteligente	-	Rendición de cuentas, open data

## 5.2 OpenRefine

### Descripción

OpenRefine es una herramienta para trabajar con datos sin procesar para limpiarlos transformándolos de un formato a otro y extendiéndolos con servicios y datos externos. Fue creado por Google y comercializado como GoogleRefine posteriormente liberando el código y pasando a tener su nombre actual [21].

La aplicación abre un servicio web local para tratar grandes cantidades de datos. La base del tratamiento es la limpieza de datos erróneos anómalos. OpenRefine trabaja con formatos comunes de texto o binarios - csv, tsv, json, xls, xml entre otros.

Los tratamientos de datos se realizan mediante filtros. La sintaxis de los filtros se puede definir en GREL, P/Jython o Clojure.

Además de filtros, se pueden realizar transformaciones sobre filas, columnas o celdas en función de los filtros anteriores.

En pos de un ejemplo, si se define el siguiente dataset:

Nombre	Apellido 1	Apellido 2	Fecha nacimiento	Género
Juan	Martínez	García	04/04/1995	H
Lucy	Ford		05/06/1996	M
Benito	Martínez	Ocasio	10-03-1994	Hombre
Pedro	Tárrega	Cola	09-01-1994	hh

Podemos observar que las últimas dos columnas de la última fila contienen datos con valores anómalos: la fecha y el género tienen un formato diferente a los de las columnas anteriores. Esto es apreciable a simple vista porque solo ejemplificamos cuatro entradas de valores, sin embargo, en un dataset de miles o millones de datos nos sería imposible. Para ello, aplicando un filtro GREL con la siguiente sintaxis sobre el campo Género:

```
length(value) < 2
```

Devolvería dos filas que cumplen los requisitos y dos que no las cumplen. Existe una gran comunidad que comparte estos filtros bajo el nombre de recetas, para extraer información de datasets comunes.

### Autoría

Proyecto originalmente creado por Metaweb Technologies, en 2010 adquirido por Google para que posteriormente en 2012 fuera convertido en un proyecto comunitario liderado por Qi Jacky Cui.

### Licencia

OpenRefine está licenciado bajo BSD-3

## Requerimientos hardware

Mínimos. La aplicación funcionaba correctamente para un grupo pequeño de usuarios, hasta alrededor de 10 usuarios, en una máquina virtual t2.micro de AWS que es equivalente a un procesador de 1 núcleo a 2.4Ghz.

## Instalación

### - Entorno

Proyecto desplegado sobre una imagen docker Ubuntu:18.04. En una máquina virtual t2.micro del servicio EC2 de AWS.

### - Dependencias

- Java, versión por defecto.
- wget/curl, versión por defecto.

### - Despliegue

Desde la raíz del sistema ejecutamos los siguientes comandos.

```
git clone https://github.com/openrefine/openrefine openrefine
cd openrefine
./refine
```

## Documentación oficial

La documentación oficial del proyecto se puede encontrar en <https://github.com/openrefine/openrefine/wiki>

## Comentarios adicionales

Se proporciona una imagen docker con OpenRefine instalado y listo para ejecutarse con los datos de ejemplo utilizados en este documento.

```
docker pull cttl/ctl:openrefine
docker run --name openrefine -it cttl/ctl:openrefine bash
cd /root/openrefine; ./refine -i 0.0.0.0
```

## Seguridad

OpenRefine provee su interfaz mediante un servidor web básico sin ningún tipo de solución de seguridad. Es recomendable aplicar algunas medidas de seguridad, tal vez del tipo proxy inverso entre otros en caso de tratar datos sensibles.



## **Valoración**

En cuanto al mantenimiento OpenRefine se puede decir que es una herramienta con una comunidad muy activa detrás. Se valoró que tenía un buen mantenimiento con unas actualizaciones muy estables que continúan a día de hoy, con más de 250 contribuidores. Actualmente en su versión 3.5.2.

La instalación es muy sencilla ya que incluye un script para un despliegue estandarizado, únicamente se necesitan conocimientos básicos para el despliegue en la nube. También resulta interesante resaltar que en su documentación oficial incluyen instaladores para los sistemas operativos Windows y MacOS.

En cuanto a su uso es de una complejidad media en comparación con otras aplicaciones ya que el objetivo de esta es el procesamiento de datos y se necesita que el usuario sepa de ello a nivel bajo al menos.

## 5.3 RocketChat

### Descripción

RocketChat es una aplicación de mensajería rápida e intuitiva que puede ayudar a cualquier equipo a mejorar sus canales de comunicación [23].

Tiene herramientas para la creación de grupos privados y de canales. Cada usuario puede también comunicarse con otro por mensajes privados. Los usuarios puede tener distintos roles para la gestión del contenido o de la configuración de la aplicación tales como “Admin” o “Propietario”.

### Autoría

Desarrollado por el grupo RocketChat.

### Licencia

Distribuido bajo la licencia MIT.

### Requerimientos hardware

Proporcionados por los desarrolladores.

Mínimo para 50 usuarios, 25 usuarios concurrentes

- Raspberry Pi 3 o Pi 2
- 4 núcleos 1 GB de memoria
- 32 GB de almacenamiento

VPS Mínimo para 200 usuarios, 50 usuarios concurrentes

- 1 núcleo (2 GHz)
- 1 GB RAM
- 30 GB de SSD

VPS Recomendado para 500 usuarios, 100 usuarios concurrentes

- 2 núcleos (2 GHz)
- 2 GB RAM
- 40 GB de SSD

## Instalación

Para probar y mostrar su funcionamiento básico recomendamos instalar la versión de docker propuesta en su guía oficial.

### - Entorno

Proyecto desplegado sobre una imagen docker Ubuntu:18.04. En una máquina virtual t2.micro del servicio EC2 de AWS.

### -Dependencias

- MongoDB 4.0.9

- NodeJS 8.11.4

Para instalar las dependencias se recomienda seguir los siguientes pasos

1. Actualizamos la lista de paquetes

```
sudo apt-get -y update
```

2. Configuramos apt para instalar los paquetes oficiales de MongoDB con la llave proporcionada comprobando que es la correcta.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv  
9DA31620334BD75D9DCB49F368818C72E52529D4
```

```
echo "deb [ arch=amd64 ] https://repo.mongodb.org/apt/ubuntu bionic/  
mongodb-org/4.0 multiverse"| sudo tee /etc/apt/sources.list.d/mongodb-  
org-4.0.list
```

3. Configuramos Node.js para que sea instalado por el manejador de paquetes.

```
sudo apt-get -y update && sudo apt-get install -y curl && curl -sL  
https://deb.nodesource.com/setup_8.x | sudo bash -`
```

4. Instalamos las herramientas de desarrollo, MongoDB, nodejs y graphicksmagick.

```
sudo apt-get install -y build-essential mongodb-org nodejs graphicksmagick
```

5. Instalamos npm si trabajamos en Ubuntu 19.04.

```
sudo apt-get install -y npm
```

6. Instalamos inherits, n y la versión que requiera Rocket.Chat, en este caso 8.11.4.

```
sudo npm install -g inherits n && sudo n 8.11.4
```

## **-Despliegue**

1. Descargamos la última versión de RocketChat

```
curl -L https://releases.rocket.chat/latest/download -o /tmp/rocket.chat.tgz
```

2. Extraemos los archivos comprimidos.

```
tar -xzf /tmp/rocket.chat.tgz -C /tmp
```

3. Instalamos las dependencias y ejecutamos RocketChat

Esta guía utiliza /opt pero puede elegir el directorio que prefiera.

```
cd /tmp/bundle/programs/server && npm install  
sudo mv /tmp/bundle /opt/Rocket.Chat
```

## **-Configuración**

1. Creamos el usuario rocketchat y le damos permisos sobre la carpeta Rocket.Chat.

```
sudo useradd -M rocketchat && sudo usermod -L rocketchat  
sudo chown -R rocketchat:rocketchat /opt/Rocket.Chat
```

2. Creamos el archivo de servicio de RocketChat.

```
cat << EOF |sudo tee -a /lib/systemd/system/rocketchat.service  
[Unit]  
Description=The Rocket.Chat server  
After=network.target remote-fs.target nss-lookup.target nginx.target mongod.  
target  
[Service]  
ExecStart=/usr/local/bin/node /opt/Rocket.Chat/main.js  
StandardOutput=syslog  
StandardError=syslog  
SyslogIdentifier=rocketchat  
User=rocketchat  
Environment=MONGO_URL=mongodb://localhost:27017  
rocketchat?replicaSet=rs01  
MONGO_OPLOG_URL=mongodb://localhost:27017/local?replicaSet=rs01
```

```
ROOT_URL=http://localhost:3000/ PORT=3000
[Install]
WantedBy=multi-user.target
EOF
```

3. Abrimos el archivo de servicio Rocket.Chat que acabamos de crear en `/usr/lib/systemd/system/rocketchat.service` y lo editamos utilizando `sudo` con el editor de preferencia para cambiar la variable `ROOT_URL` por la URL que quieras usar para acceder al server (por defecto estará `http://localhost:3000` como URL de acceso).

Opcionalmente se puede cambiar también `MONGO_URL`, `MONGO_OPLOG` y `PORT`.

```
MONGO_URL=mongodb://localhost:27017/rocketchat?replicaSet=rs01
MONGO_OPLOG_URL=mongodb://localhost:27017/local?replicaSet=rs01
ROOT_URL=http://your-host-name.com-as-accessed-from-internet:3000
PORT=3000
```

4. Configuraremos el almacenamiento de la base de datos (storage engine) y la replicación para MongoDB.

```
sudo sed -i "s/^# engine:/ engine: mmapv1/" /etc/mongod.conf
sudo sed -i "s/^#replication:/replication:\n replSetName: rs01/" /etc/mongod.conf
```

5. Habilitamos MongoDB y ejecutamos RocketChat.

```
sudo systemctl enable mongod && sudo systemctl start mongod
mongo --eval "printjson(rs.initiate())"
sudo systemctl enable rocketchat && sudo systemctl start rocketchat
```

6. Tendremos nuestro servicio desplegado en `http://el_nombre_de_tu_url_o_tu_ip:3000`

## Documentación oficial

→ RocketChat <https://es.rocket.chat/>

→ Guía de instalación <https://es.rocket.chat/install>

## Seguridad

Recomendamos configurar el servicio como `https`, cambiar la URL de acceso por defecto y configurar el control de acceso a la base de datos.

## **Valoración**

RocketChat es una aplicación de mensajería con canales, grupos y roles, es perfecta para cualquier tamaño de organización.

RocketChat recibe constantes actualizaciones y cambios, tiene un mantenimiento constante. Lleva varios años de crecimiento y tiene contribuciones diarias con casi 800 contribuidores. Su última versión estable es la 4.7.0.

La instalación es de complejidad media pues hay que instalar y configurar varias librerías o frameworks previos ya que la aplicación tiene parte de backend, frontend y de base de datos. Su configuración también requiere de varios pasos que pueden resultar algo confusos para un usuario que no conozca este ámbito técnico. Por otro lado estos pasos están altamente probados por una gran cantidad de usuarios y los más de 7.400 “forks” de la aplicación avalan la estabilidad de su despliegue.

El uso de la aplicación es bastante sencillo ya que se su funcionalidad principal es la de mensajería y su interfaz, clara y sencilla, se asemeja a otras del tipo Slack o Microsoft Teams.

## 5.4 FixMyStreet

### Descripción

FixMyStreet es una herramienta de recolección e informe de problemas que pueden surgir en una calle - una farola estropeada, un semáforo que no funciona, etc. FMS define órganos - ayuntamientos, concejalías, etc - a los que se pasarán los reportes creados a partir de los problemas de los que los usuarios han ido informando utilizando el servicio FMS. Los usuarios pueden informar de un problema creándose una cuenta de usuario para configurar sus propios informes en cualquier otro momento, o hacerlo de manera anónima. Asimismo, cada informe permite recoger un título, una descripción del problema y varias fotos [18].

En cuanto a la comunicación con los órganos oficiales, esta es transparente al usuario, sin embargo, a la hora de configurar un órgano, se tiene que establecer qué tipo de envío se realizará, teniendo como opciones Open311 o email. Se recomienda esta última ya que la primera está más dirigida a los ciudadanos de Estados Unidos como se explica en la sección "Tecnologías utilizadas".

A la hora de administrar el servicio, la interfaz /admin ofrece una gran variedad de ajustes. En primer lugar y como se ha explicado anteriormente, la configuración de órganos oficiales a los que mandar los informes de los usuarios. Dado un órgano, se pueden establecer sobre el mismo Plantillas, Prioridades y Estados, así como administrar usuarios - creación y eliminación -

### Autoría

Servicio desarrollado y mantenido por MySociety

### Licencia

Servicio licenciado bajo GNU AGPLv3.

### Requerimientos hardware

Una instalación en un entorno de test o desarrollo no necesitará gran capacidad para correr el servicio. Desglosando los requerimientos según estándares cloud/VPS, los servidores con características mínimas de los principales proveedores son suficientes. En AWS serviría una instancia t2.micro. Sin embargo, en entornos de producción la medida aumenta en función del alcance de los usuarios así de qué otras herramientas se empotran dentro del servicio.

### Instalación

#### - Entorno

Proyecto desplegado sobre una imagen docker Ubuntu:18.04. En una máquina virtual t2.micro del servicio EC2 de AWS.

## - Dependencias

- postgresql 10
- wget

## - Despliegue

```
wget https://raw.githubusercontent.com/mysociety/commonlib/master/bin/install-site.sh
```

Suponemos que el servicio postgres está iniciado y configurado. Si no es así, a continuación se propone una configuración básica de este servicio.

```
vim/vi/nano conf/general.yml
```

Editamos el fichero de configuración general con los siguientes datos:

```
{
  # PostgreSQL database details for FixMyStreet
  FMS_DB_HOST: ""
  FMS_DB_PORT: '5432'
  FMS_DB_NAME: 'fixmystreet'
  FMS_DB_USER: 'postgres'
  FMS_DB_PASS: 'postgres'
}
```

Reconstruimos el esquema en la base de datos e instalamos:

```
cd /home/fms/fixmystreet
```

```
./script/setup
```

Los siguientes comandos instalarán el servicio y crearán la carpeta fixmystreet dentro de \$HOME del nuevo usuario fms con el dominio fixmystreet.local.

```
sh install-site.sh fixmystreet fms fixmystreet.local
```

```
su - fms
```

```
cd fixmystreet
```

```
script/server
```

Llegados a este punto, el servicio debe estar disponible en el puerto 3000 de la máquina donde se instaló.

## Documentación oficial

→ Guía de instalación oficial <https://fixmystreet.org/install/>



## **Seguridad**

El script de instalación configura automáticamente un proxy inverso con Nginx para manejar el servicio FMS por el puerto 80, sin encriptación, en un fichero de configuración con el nombre del dominio dado en la instalación - /etc/nginx/sites- available/fixmystreet.local. Recomendamos, como medida de seguridad si los datos a tratar fueran de carácter sensible, crear un certificado gratuito de Let's Encrypt, realizar una configuración de Nginx parecida a la de por defecto pero utilizando el puerto 443 y redireccionar todas las peticiones del puerto 80 al anterior.

## **Valoración**

La instalación de Fixmystreet puede ser relativamente compleja en función de lo que se necesite, esto es, el tiempo necesario para una instalación rápida en entornos de test o de proyectos pequeños puede ser bastante corto, sin embargo, una instalación en un entorno de producción y adaptada a una organización concreta puede ser compleja por la cantidad de servicios a los que se puede unir este proyecto. Por otro lado, gracias al reconocimiento y la gran cantidad de implementaciones en grandes entornos, se trata de un servicio muy robusto. La aplicación es fácil de usar e incluye una aplicación para móviles para reportar al servicio web desplegado en el proyecto.

Actualmente Fixmystreet tiene un gran mantenimiento y actualizaciones constantes de sus 50 contribuidores por lo que se considera una herramienta estable para su uso en proyectos cotidianos.

## 5.5 Kanboard

### Descripción

Kanboard es una aplicación para la gestión de tareas de un proyecto basado en el sistema de información kanban [19].

Kanban es un sistema de trabajo que está ampliamente adoptado en muchas de las empresas tecnológicas e industriales actuales. Nació como evolución del sistema de producción desarrollado por Toyota alrededor de 1939 conocido como “Just-in-time”. Este sistema tenía como objetivo reducir los tiempos de espera y el sobrestock en cadenas de montaje donde por falta de una buena planificación y flujos de trabajo se volvían ineficientes sobretodo a principio de mes donde partes de la cadena estaban simplemente paradas a la espera de procesos anteriores. El sistema buscaba fabricar únicamente lo necesario en el momento necesario entregando solo las piezas y componentes que fueran estrictamente necesarias en las cadenas de montaje de ese mismo día. Este proceso no se pudo implantar correctamente hasta 1954 y mejoró la eficacia y la eficiencia de las plantas donde se adoptó.

El método Kanban nace de la base que creó el “Just-in-time” y ayuda a mantener una visualización total de la cadena de trabajo. En este sistema se dividen los objetivos en tareas atómicas y estas se sitúan en diferentes columnas que representan los diferentes estados que tendrá una tarea antes de ser completada. En la imagen siguiente se muestra la interfaz de Kanboard donde se aprecian estos estados [5].

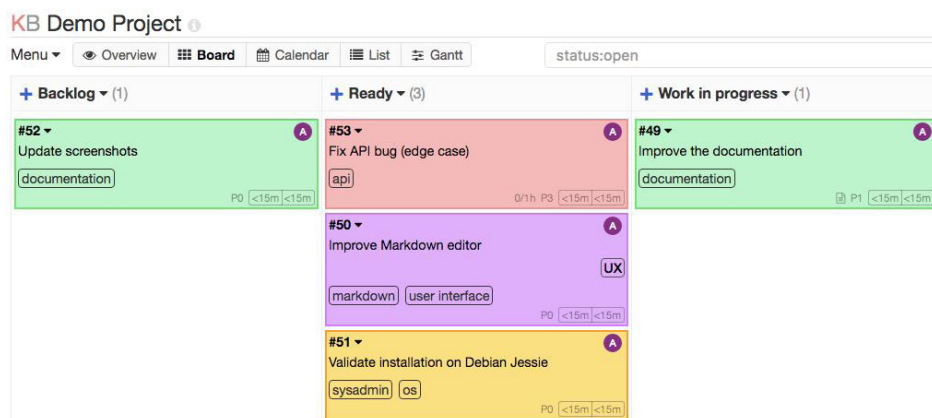


Imagen 1. Interfaz de Kanboard

### Autoría

Principalmente desarrollado por Frédéric Guillot.

### Licencia

Está distribuido bajo licencia del MIT.

### Requerimientos hardware

Para una presentación o un testeo de la aplicación, para aproximadamente 5 usuarios concurrentes, no necesitará más de 1-2 GB de RAM y 1-3 GB de almacenamiento pero conforme aumenten el número de usuarios y accesos esta configuración será insuficiente.

## Instalación

### -Entorno

Proyecto desplegado sobre una imagen docker Ubuntu:18.04. En una máquina virtual t2.micro del servicio EC2 de AWS.

### -Dependencias

- Apache2
- PHP (cli, sqlite3, opcache, json, mysql, pgsql, ldap, xml)

Estos son los pasos de instalación de las dependencias PHP que se necesitarán para la instalación de Kanboard.

Actualizamos la lista de paquetes.

```
sudo apt-get -y update
```

Añadimos el repositorio siguiente

```
sudo add-apt-repository ppa:ondrej/php
```

Instalamos apache2 y las versiones de las librerías que necesitemos para trabajar con PHP y nuestra base de datos. Descargamos las versiones 7.1 ya que la 7.0 ya no está disponible.

```
sudo add-apt-repository ppa:ondrej/php
```

```
sudo apt-get install -y apache2 libapache2-mod-php7.1 php7.1-cli php7.1-mbstring php7.1-sqlite3 \ php7.1-opcache php7.1-json php7.1-mysql php7.1-pgsql php7.1-ldap php7.1-gd php7.1-xml
```

### -Despliegue

Para probar y mostrar su funcionamiento básico recomendamos instalar la versión docker propuesta en su cuenta. ([https://docs.kanboard.org/en/latest/admin\\_guide/docker.html](https://docs.kanboard.org/en/latest/admin_guide/docker.html))

A continuación descargamos la última versión de Kanboard desde su repositorio oficial. Comprobamos cual es su última versión. En nuestro caso es la 1.2.9.

```
cd /var/www/html
```

```
wget https://github.com/kanboard/kanboard/archive/v1.2.9.zip
```

Descomprimos el fichero que hemos descargado (añadimos el comando para instalar unzip en el caso de no tenerlo instalado).

```
sudo apt-get install unzip
```

```
unzip v1.2.9.zip
```

Le damos permisos sobre la carpeta que acabamos de descomprimir y eliminamos el archivo comprimido.

```
chown -R www-data:www-data kanboard-&lt;version&gt;/data
```

```
rm v1.2.9.zip
```

La aplicación estará escuchando en el puerto 80 y podremos acceder a ella desde la URL <http://localhost/kanboard-1.2.9/>.

### **Documentación oficial**

→ Kanboard <https://kanboard.org/>

→ Guía de instalación oficial

[https://docs.kanboard.org/en/latest/admin\\_guide/installation.html](https://docs.kanboard.org/en/latest/admin_guide/installation.html)

### **Comentarios adicionales**

Recomendamos utilizar esta guía para la instalación en Ubuntu 18.04 y PHP 7.1 en adelante puesto que la oficial utiliza una versión de Ubuntu y PHP más antigua.

### **Seguridad**

Recomendamos configurar el servicio como https, cambiar el usuario y contraseña por defecto y controlar el acceso al directorio data y a otros ficheros de configuración del servidor y la base de datos de vuestra elección.

### **Valoración**

La aplicación Kanboard tiene una instalación y despliegue bastante sencillo y no se requieren conocimientos adicionales para su configuración. Es de uso sencillo y cuenta con una interfaz intuitiva muy parecida a la de otras aplicaciones de gestión de un sistema Kanban como Jira o Trello.

Kanboard tiene más de 300 colaboradores pero se encuentra en una fase de mantenimiento. La aplicación sigue recibiendo cambios pero no tiene abiertos desarrollos de funcionalidades nuevas.

## 5.6 LimeSurvey

### Descripción

LimeSurvey es una herramienta de código abierto que permite hacer y gestionar diferentes tipos de encuestas online. Está diseñada para que su uso habitual pueda ser cómodo desde móvil, tablet y ordenador [20].

El uso de LimeSurvey es relativamente sencillo y este pone a disposición del usuario diferentes plantillas que permiten añadir complejidad al cuestionario como por ejemplo preguntas dinámicas en base a respuestas anteriores o saltos entre preguntas en diferentes partes de un mismo formulario.

Algunos ejemplos de encuestas que ofrece la herramienta son:

- Estudios de mercado.
- Calificación de proveedores.
- Encuesta de opinión general.
- Votaciones.
- Encuestas de satisfacción de cliente.
- Encuestas internas de empleados.
- Encuestas de pie de calle para organizaciones benéficas.

Para probar y mostrar su funcionamiento se recomienda visitar la demo oficial como administrador (<https://demo.limesurvey.org/index.php?r=admin>) para crear y gestionar encuestas o probar a rellenar una en la encuesta de prueba (<https://survey.limesurvey.org/78184?lang=en>) propuesta en el proyecto.

### Autoría

Desarrollado por el equipo de LimeSurvey.

### Licencia

Está distribuido bajo la licencia GPL 2.0.

### Requerimientos hardware

Para una presentación o un testeo mínimo de la aplicación no necesitará mas de 1 núcleo virtual, 1 o 2 GB de RAM y 250 MB de almacenamiento pero conforme aumenten el número de usuarios será necesario aumentar la RAM y la capacidad de almacenamiento.

## Instalación

### - Entorno

Esta aplicación fue instalada en una instancia t2.micro del servicio de Amazon de cloud computing (AWS) con Ubuntu 18.04.

### -Dependencias

- Apache >= 2.4 | nginx >= 1.1 | otra versión de servidor web con PHP
- php >= 5.4 con mbstring y los drivers pdo-database
- mysql >= 5.5.3 | pgsql >= 9 | mariadb >= 5.5 | mssql >= 2005
- Tener habilitadas las librerías por defecto de PHP.

### -Recomendaciones

- nginx 1.4.6
- php 5.6.x with php-fpm, mbstring, gd2 with freetype, imap, ldap, zip, zlib and database drivers (muy recomendado)
- mysql 5.5.50

### -Despliegue

En este caso vamos a instalar la versión de servidor.

Descargamos la última versión de LimeSurvey y la descomprimos en el directorio de nuestro servidor web.

```
unzip limesurvey3.17.5+190604.zip
```

Ahora tendremos que permitir la lectura y escritura de algunos directorios por nuestro servidor web. Los directorios son:

```
"/limesurvey/tmp"
```

```
"/limesurvey/upload/"
```

```
"/limesurvey/application/config/"
```

Si se utiliza Apache, se puede hacer uso de los siguientes comandos para permitir la lectura por todos pero la escritura solo por el servidor web.

```
chmod -R 755 <directory>
```

```
chown -R apache <directory>
```

Lo siguiente que tendremos que hacer es crear un usuario de la base de datos que estemos utilizando y opcionalmente una base de datos para la aplicación (en caso de no crearla ahora se creará automáticamente durante el proceso de instalación). Se recomienda que el usuario tenga los siguientes permisos o equivalentes (ejemplo de Mysql): SELECT, CREATE, INSERT, UPDATE, DELETE, ALTER, DROP, INDEX.

Si utiliza Mysql puede crear un usuario de la siguiente forma

```
mysql -u root -p
CREATE USER 'nombre_usuario'@'localhost' IDENTIFIED BY 'tu_contrasena'
GRANT [permiso] ON [nombre de bases de datos].[nombre de tabla] TO '[nombre de usuario]'@'localhost';
```

Iremos ahora a "localhost/limesurvey/admin" para acceder al script de instalación. Aquí se nos preguntará por el usuario que acabamos de crear y si queremos crear una nueva base de datos.

**IMPORTANTE:** si estamos instalando el servicio en una máquina EC2 de AWS probablemente no tengamos facilidades para ejecutar el script de instalación desde un navegador. Si se quiere acceder al script de instalación desplegado en la máquina EC2 desde local tenga en cuenta que deberá configurar la conexión remota con la base de datos (para el caso de Mysql: habilitar en my.conf la ip de acceso, el puerto 3306 en el servidor, etc...).

Una vez terminado el proceso de instalación podrá ir a "http://localhost:puerto\_elegido/limesurvey/admin". Si no hemos cambiado las credenciales durante la instalación podremos acceder mediante las credenciales por defecto.

```
User: admin
Password: password
```

## Documentación oficial

→ LimeSurvey (<https://github.com/LimeSurvey/LimeSurvey>)

## Seguridad

Recomendamos configurar el servicio como https, cambiar la URL de acceso por defecto y controlar el acceso a la base de datos.

## Valoración

LimeSurvey es una aplicación muy extendida y relativamente fácil de usar tanto por parte del administrador como por parte del cliente al que se encueste. Es intuitiva pero compleja, tiene bastantes opciones que pueden complicar un poco su uso al principio. Es bastante flexible y permite mostrar los datos obtenidos de forma gráfica.

La instalación y despliegue es de complejidad media ya que conlleva el despliegue de una base de datos además de su configuración. Esta complejidad se reduce si el equipo donde se instale tiene una interfaz gráfica ya que la aplicación tiene un script de instalación que se ejecuta en el navegador y ayuda a agilizar el proceso de instalación.

LimeSurvey tiene actualizaciones constantes y casi 200 colaboradores. Es una aplicación robusta y estable.



## 5.7 RAWGraphs

### Descripción

RAWGraphs es una herramienta web abierta que te permite visualizar mediante gráficos datos introducidos en forma de tabla o texto. Es el punto de unión entre tablas de datos y distintas formas de visualización. La aplicación cuenta con decenas de formas de representación y permite crear también las necesarias que se adapten a la necesidad del usuario. Las visualizaciones creadas por la aplicación son fácilmente exportables a diferentes formatos como .svg y .png. En su página oficial se pueden probar diferentes datasets que proporcionan en un entorno de pruebas [22]. La ejecución es mediante el navegador web por lo que no se ejecuta o se guarda ninguna información por parte del servidor.

### Autoría

Desarrollado por Giorgio Caviglia, Michele Mauri, Giorgio Uboldi y Matteo Azzi. Mantenido por DensityDesign Research Lab y Calibro.

### Licencia

Está distribuido bajo la licencia Apache 2.0.

### Requerimientos hardware

Para una presentación o un testeo de la aplicación no necesitará más de 1 núcleo virtual, 1 o 2 GB de RAM y menos de 1 GB de almacenamiento pero conforme aumenten el número de usuarios es recomendable aumentar la RAM y prestaciones del procesador.

### Instalación

#### - Entorno

Esta aplicación fue instalada en una instancia t2.micro del servicio de Amazon de cloud computing (AWS) con Ubuntu 18.04.

#### - Dependencias

- Git
- Bower
- Npm

## **-Despliegue**

Clonamos RAWGraphs.

```
sudo git clone https://github.com/densitydesign/raw.git
```

Nos movemos al fichero raíz de RAWGraphs.

```
cd raw
```

Instalamos npm y bower utilizando npm.

```
sudo apt install npm build-essential
```

```
npm install -g bower
```

Añadimos el script analytics.js.

```
cp js/analytics.sample.js js/analytics.js
```

Ahora podemos ejecutar RAWGraphs en este caso utilizaremos python 3+ y lo pondremos a escuchar en el puerto 4000.

```
python -m http.server 4000
```

Ahora podremos acceder a la aplicación en <http://localhost:4000/>.

## **Documentación oficial**

→ RAWGraphs (<https://github.com/densitydesign/raw/blob/master/README.md>)

## **Seguridad**

Recomendamos configurar el servicio como https y cambiar la URL de acceso por defecto.

## **Valoración**

RAWGraphs es una aplicación que sirve como enlace entre plantillas y tablas de cálculo y diferentes modelos gráficos. Admite diferentes tipos de formatos de entrada y tiene una gran variedad de tipos de visualizaciones. Es una aplicación útil para mostrar información y flexible para introducirse en una aplicación web más compleja. Disponen de una versión de prueba ya desplegada en <https://app.rawgraphs.io/> para su uso abierto. La aplicación se ejecuta íntegramente en el navegador y no tiene operaciones en el lado del servidor o se guardan datos sobre estas.

Tiene una instalación, configuración y despliegue sencillos ya que no tiene muchas dependencias. La aplicación recibe cambios constantes aunque tiene pocos colaboradores y su uso está más o menos extendido.

## 5.8 Demokratian

### Descripción

Demokratian es una aplicación de código abierto de colaboración ciudadana que permite hacer distintos tipos de votaciones online y presenciales. Esta aplicación proporciona una plataforma de votaciones web que permite a distintos grupos de ciudadanos realizar consultas entre sus participantes permitiendo una forma de decisión horizontal en la comunidad en la que se implemente [17].

La herramienta permite hacer votaciones simultáneas que tienen en cuenta en tiempo real las votaciones de cada uno de sus participantes permitiendo que pueda controlar votaciones online y presenciales mixtas ya que evita un “doble voto” si alguien vota online y luego intenta votar presencialmente. Si se habilita proporciona también anonimidad para las votaciones que lo requieran y se puede restringir las votaciones por censos.

En el caso de las votaciones mixtas (presencial y online) no se podría dar el caso de un doble voto fraudulento con la misma identificación puesto que antes de poder votar se comprueba en tiempo real la identificación de la persona con la identificación asociada a esta en el censo de la votación.

Demokratian ofrece cuatro tipos de votación.

**De voto ponderado.** En este tipo de votación el usuario tendrá una lista de opciones en una columna a la izquierda que tendrá que ordenar en una segunda columna a la derecha. El cálculo de la votación es ponderado y puede hacerse de formas. Mediante el método Dowdall donde la primera opción recibirá 1 punto, la segunda 0.5 puntos, la tercera 0.25 y así sucesivamente. O por el método borda dónde el último en la lista tendrá 1 punto, el penúltimo 2 puntos, el antepenúltimo 3 puntos y así sucesivamente.

**De Voto Único Transferible (VUT).** Es una modalidad de voto por orden de preferencia. Mediante este tipo de votación el usuario ordena una lista de propuestas o candidatos y si, por ejemplo, tu candidato preferido o primera opción ya ha salido elegido o es eliminado tu voto pasaría a la siguiente opción de la lista. Este sistema busca minimizar el efecto del “voto útil” ya que se pierde la incertidumbre de votar propuestas o candidatos que al final puedan no tener suficientes apoyos. Este tipo de votación ha sido adoptado en países como Irlanda o Malta para las elecciones de sus representantes políticos.

**De tipo encuesta.** Esta es la modalidad más sencilla de votación donde se permite elegir una o más opciones por cada pregunta que se añada. No tiene ningún tipo de ponderación.

**De tipo debate.** En este sistema se puede votar de manera dinámica mientras ocurre una conversación mediante mensajes que se publican en un tablón al estilo foro. El usuario puede cambiar de opinión en cualquier punto del debate y cambiar su voto en consecuencia. Al final del debate las votaciones se cierran y se bloquea el cambio de voto. Este sistema no asegura la trazabilidad del voto ya que mantiene el estado de la votación en la base de datos durante el debate y alguien con acceso a esta podría llegar a ver qué ha votado un usuario en concreto.

## **Autoría**

Desarrollado principalmente por Carlos Salgado y el equipo de Demokratian.

## **Licencia**

Está distribuido bajo la licencia GPL 3.0.

## **Requerimientos hardware**

Para una presentación o un testeo mínimo de la aplicación podrá utilizar un tipo de instancia t2.micro, equivalente a 1 núcleo y 1 o 2 GB de RAM. Dependiendo del número de usuarios tendrá que aumentar bastante esta especificación.

## **Instalación**

### **- Entorno**

Esta aplicación fue instalada en una instancia t2.micro del servicio de Amazon de cloud computing (AWS) con Ubuntu 18.04.

### **- Dependencias**

- Apache
- PHP >= 7
- Base de datos mysql

### **-Despliegue**

Hay dos formas de instalar Demokratian, la manual y la automática. En este caso hemos instalado la automática.

Primero tendremos que subir todos los archivos ([https://bitbucket.org/csalgadow/demokratian\\_votaciones/src/master/](https://bitbucket.org/csalgadow/demokratian_votaciones/src/master/)) de la aplicación a nuestro servidor. Esto se puede hacer mediante el comando git clone desde nuestra máquina virtual.

```
Git clone https://bitbucket.org/csalgadow/demokratian_votaciones/src/master/
```

Tendremos que darle permisos de escritura al usuario con el que estemos utilizando Apache en las siguientes carpetas: config, upload\_user, upload\_pic, data\_vut. Para ello podemos usar el comando chmod 700 en esas carpetas.

```
chmod 700 ./config
```

Lo siguiente que se tendrá que hacer es ir a la carpeta install utilizando nuestro navegador web ([midominio.org/install](http://midominio.org/install)) y seguir los pasos de instalación que se mostrarán. Dado que tendrás que proporcionar datos de configuración de la base de datos y del servidor de correo recomendamos configurar ambos antes de este paso.

**IMPORTANTE:** si estamos instalando el servicio en una máquina EC2 de AWS probablemente no tengamos facilidades para ejecutar el script de instalación desde un navegador. Si se quiere acceder al script de instalación desplegado en la máquina EC2 desde local tenga en cuenta que deberá configurar la conexión remota con la base de datos (para el caso de Mysql habilitar en my.conf la ip de acceso, el puerto 3306 en el servidor, etc...).

### **Documentación oficial**

→ Demokratian  
([https://bitbucket.org/csalgadow/demokratian\\_votaciones/wiki/Instalaci%C3%B3n](https://bitbucket.org/csalgadow/demokratian_votaciones/wiki/Instalaci%C3%B3n))

### **Seguridad**

Recomendamos configurar el servicio como https, cambiar la URL de acceso por defecto y controlar el acceso a la base de datos. También se recomienda aumentar la seguridad y restringir los canales de acceso a la base de datos en el caso de la modalidad de votación de tipo debate ya que esta por su naturaleza no asegura la no trazabilidad del voto y alguien con acceso a la base de datos podría las votaciones de los usuarios.

### **Valoración**

Demokratian es una aplicación fácil de instalar en local aunque aumenta su complejidad en despliegues en la nube puesto que normalmente no se tiene acceso a la interfaz gráfica que se necesita para el script que facilita la instalación y configuración de la aplicación. En este caso la complejidad radicaría en la configuración de una conexión remota a la base de datos.

La aplicación ha cambiado su versión 4 en el último año y tiene actualizaciones constantes de su colaborador principal y creador Carlos Salgado.

Esta herramienta probablemente sea de las más concretas en cuanto a su finalidad, es muy potente y abre camino a votaciones más seguras y colaborativas. Los requerimientos hardware de esta aplicación no son muy altos al principio pero necesitan escalar rápidamente para grupos grandes de usuarios simultáneos.

## 6. Conclusiones

Gracias al proyecto Civic Tech Test Lab se han podido clasificar y dar a conocer once aplicaciones de las cuales en esta memoria se han incluido las siete desplegadas y documentadas por el autor de este trabajo. Se ha cumplido el objetivo de crear un repositorio de herramientas libres útiles para la gestión de grupos de ciudadanos que quieran tener mayor autonomía y soberanía tecnológica. El proyecto también ayuda a dar visibilidad y acercar proyectos de código libre a comunidades que puedan serles de utilidad. Este desarrollo ha servido también como nexo de unión entre necesidades tecnológicas y las herramientas libres que pueden resolverlas, gestionando el descubrimiento, investigación, pruebas y documentación necesarias.

Como parte del esfuerzo del proyecto para acercar este desarrollo la documentación del repositorio se puede encontrar en castellano, valenciano e inglés. Para facilitar también cualquier contribución el repositorio se encuentra público con una licencia libre.

Para que fuera posible el desarrollo de este proyecto se han utilizado conocimientos aprendidos en diferentes asignaturas de la carrera de Ingeniería Informática de la Universidad Politécnica de Valencia.

Han sido especialmente útiles las asignaturas de Tecnologías de Sistemas de Información en Red (TSR) y Desarrollo Web (DEW) ya que ambas te introducen al lenguaje JavaScript que ha sido predominante en las aplicaciones documentadas y la primera además te introduce a la tecnología Docker que ha sido muy útil conocer para el despliegue rápido de muchas de las herramientas analizadas.

También se han tenido que conocer aunque sea a nivel básico varios tipos de lenguajes de programación y tanto bases de datos relacionales como no relacionales por lo que las asignaturas de Lenguajes, Tecnologías y Paradigmas de la programación (LTP) como Bases de Datos (BDA) han sido especialmente útiles. Tener los conocimientos sobre interfaces proporcionados por las asignaturas de Interfaz persona-computador (IPC) y Desarrollo Centrado en el Usuario (DCU) también ha facilitado hacer un análisis básico de la usabilidad de las herramientas que desplegamos.

Quiero dedicar el final de este capítulo a hablar brevemente de los Objetivos de Desarrollo Sostenible (ODS) y la relación de algunos de ellos con este trabajo. Los ODS son unos objetivos promovidos por La Organización de las Naciones Unidas (ONU) que nacieron en 2015 con la idea de ampliar los Objetivos de Desarrollo del Milenio (ODM). Existen diecisiete objetivos y entre ellos hay objetivos que se centran en el cambio climático, la pobreza, igualdad de género, la salud, la paz... Entre muchas otras cosas. Este trabajo y las herramientas expuestas tienen también un carácter social, puesto que abogan por la soberanía tecnológica, el uso responsable de la tecnología y la libertad de esta. Por ello creo que están directamente relacionados con los objetivos "Industria, innovación e infraestructura" y "Reducción de las desigualdades" por ayudar a que la innovación tecnológica sea libre, cercana y para todos.

En cuanto al objetivo "Industria, innovación e infraestructura" los proyectos de tecnologías libres ayudan a impulsar industrias y negocios a nivel mundial. Hacer más accesible

herramientas digitales facilita el desarrollo de todos los sectores de cualquier país. Además crean un fondo de conocimiento global del que puede beneficiarse y contribuir quien quiera facilitando la creación de infraestructuras digitales en países en vías de desarrollo. Dentro de este objetivo una de las metas a la que mas contribuyen estos movimientos es la de aumentar el acceso a tecnologías de la información, comunicaciones y de internet puesto que el objetivo del soberanía tecnológica es democratizar y hacer accesible el conocimiento aplicado a la tecnología.

Esto último también enlaza con el objetivo de “Reducción de las desigualdades” puesto que tanto los movimientos del código libre y de la soberanía digital buscan reducir las desigualdades entre comunidades haciendo que los avances y soluciones tecnológicas no sean para unos pocos si no para el avance y disfrute de todos, sin importar ninguna condición política o personal.

# 7. Bibliografía

## Artículos, libros y tesis

[1] CALDERÓN BELTRÁN, Natalia. *Technological sovereignty: What chances for alternative practices to emerge in daily IT use?*. Hybrid. Revue des arts et médiations humaines, 2016, no 3.

[2] CORBLY, James Edward. *The Free Software Alternative: Freeware, Open Source Software, and Libraries*. Information Technology and Libraries, 2014, vol. 33.

[3] HACHÉ, Alex. *Technological Sovereignty. Passarelle*, For Free Information and Open Internet. 2014, 11.

[4] HERON, Michael; HANSON, Vicki L.; RICKETTS, Ian. *Open source and accessibility: advantages and limitations*. Journal of interaction Science, 2013, vol. 1, no 1.

[5] JAPAN MANAGEMENT ASSOCIATION, et al. *KANBAN: Y Just-in-time en Toyota*. Routledge, 2018.

[6] LÓPEZ VELARDE, Jesús Armando. *Derechos de la soberanía digital*. México: La Biblioteca, 2021. \*

[7] STALLMAN, Richard. *Software libre para una sociedad libre*. Madrid, Spain: Traficantes de Sueños, 2004. ISBN 9788493355517.

[8] STALLMAN, Richard. *Why Free Software Is More Important Now Than Ever Before*. Passarelle, For Free Information and Open Internet. 2014, 11.

[9] VIDAL MONTBLANCH, Omayra. *Empoderamiento ciudadano en Benimaclet: autogestión y movimientos sociales*. Trabajo de fin de grado, Universidad Politécnica de Valencia, 2020.

## Páginas web

[10] A collaborative model and open standard for civic issue tracking. *Open311* [en línea]. Disponible en: <https://www.open311.org/>

[11] Discover popular open source alternatives to your proprietary SaaS. *Open Source Alternatives* [en línea]. Disponible en: <https://www.opensourcealternative.to/>

[12] Open Source Initiative. Disponible en: <https://opensource.org/> \*

[13] Open source, dependencies, and licensing by the people at FOSSA. *Fossa* [en línea]. Disponible en: <https://fossa.com/blog/>



[14] Projects and Applications. *Opensource* [en línea]. Disponible en: <https://opensource.com/resources/projects-and-applications>

[15] The greatest collection of open source and free software. *Open Source Software Directory* [en línea]. Disponible en: <https://opensourcesoftwaredirectory.com/>

## Repositorios

[16] Civic technologies compilation. *GitHub* [en línea]. Disponible en: <https://github.com/cttlrepository/cttl>

[17] Demokratian votaciones. *Bitbucket* [en línea]. Disponible en: <https://bitbucket.org/csalgadow/demokratian-v4/>

[18] Fixmystreet: This is mySociety's popular map-based reporting platform: easy to install in new countries and regions. *GitHub* [en línea]. Disponible en: <https://github.com/mysociety/fixmystreet>

[19] Kanboard: Kanban project management software. *GitHub* [en línea]. Disponible en: <https://github.com/kanboard/kanboard>

[20] LimeSurvey: The most popular FOSS online survey tool on the web. *GitHub* [en línea]. Disponible en: <https://github.com/LimeSurvey/LimeSurvey>

[21] OpenRefine: Free, open source power tool for working with messy data and improving it. *GitHub* [en línea]. Disponible en: <https://github.com/OpenRefine/OpenRefine>

[22] Rawgraphs: A web interface to create custom vector-based visualizations on top of RAWGraphs core. *GitHub* [en línea]. Disponible en: <https://github.com/rawgraphs/rawgraphs-app>

[23] Rocket.Chat: The communications platform that puts data protection first. *GitHub* [en línea]. [sin fecha]. Disponible en: <https://github.com/RocketChat/Rocket.Chat>

# 8. Anexos

## 8.1. Motivación Personal

Encontré que este proyecto me motivaba de dos maneras: una profesional y otra más personal. La profesional estaba relacionada con los conocimientos de despliegue de infraestructuras cloud y los procesos e integraciones de desarrollos de código libre.

Por la parte más personal, siempre me ha atraído el software libre ya que fue gracias a librerías, cursos y herramientas libres que pude acercarme al mundo de la tecnología y de verdad encontrar a lo que me quería dedicar. Esta manera de pensar me atrajo mucho hacia la TCE que descubrí gracias a que estaban muy inmersos en diferentes proyectos libres en colaboración con diferentes estudiantes de la ETSINF.

Durante un tiempo previo a la publicación de esta beca estuve siguiendo y atendiendo a diversos cursos y reuniones de la cátedra que hicieron afianzar mi idea de trabajar en una iniciativa suya. La filosofía de sus proyectos estaba claramente plasmada en su organización interna y noté que allí se trabajaba con mucho cariño, energía y pasión.

Actualmente utilizo incontables recursos libres en el día a día de mi trabajo que me ayudan a desarrollar herramientas de manera más rápida y efectiva.

## 8.2. Relación del trabajo con los Objetivos de Desarrollo Sostenible

El 25 de septiembre de 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.

Los ODS son unos objetivos promovidos por La Organización de las Naciones Unidas (ONU) que nacieron con la idea de ampliar los Objetivos de Desarrollo del Milenio (ODM). Existen diecisiete objetivos y entre ellos hay objetivos que se centran en el cambio climático, la pobreza, igualdad de género, la salud, la paz...

1. Fin de la pobreza
2. Hambre Cero
3. Salud y Bienestar
4. Educación de Calidad
5. Igualdad de género

6. Agua limpia y saneamiento
7. Energía asequible y no contaminante
8. Trabajo decente y crecimiento económico
9. Industria innovación e infraestructura
10. Reducción de las desigualdades
11. Ciudades y comunidades sostenibles
12. Producción y consumos responsables
13. Acción por el clima
14. Vida submarina
15. Vida de ecosistemas terrestres
16. Paz, justicia e instituciones
17. Alianzas para lograr objetivos.

Este trabajo y las herramientas expuestas tienen también un carácter social, puesto que abogan por la soberanía tecnológica, el uso responsable de la tecnología y la libertad de esta. Por ello creo que están directamente relacionados con los objetivos “Industria, innovación e infraestructura” y “Reducción de las desigualdades” por ayudar a que la innovación tecnológica sea libre, cercana y para todos.

En cuanto al objetivo “Industria, innovación e infraestructura” los proyectos de tecnologías libres ayudan a impulsar industrias y negocios a nivel mundial. Hacer más accesible herramientas digitales facilita el desarrollo de todos los sectores de cualquier país. Además crean un fondo de conocimiento global del que puede beneficiarse y contribuir quien quiera, facilitando la creación de infraestructuras digitales en países en vías de desarrollo. Dentro de este objetivo una de las metas a la que más contribuyen estos movimientos es la de aumentar el acceso a tecnologías de la información, comunicaciones y de internet puesto que el objetivo de la soberanía tecnológica es democratizar y hacer accesible el conocimiento aplicado a la tecnología.

Esto último también enlaza con el objetivo de “Reducción de las desigualdades” puesto que tanto los movimientos del código libre y de la soberanía digital buscan reducir las desigualdades entre comunidades haciendo que los avances y soluciones tecnológicas no sean para unos pocos si no para el avance y disfrute de todos, sin importar ninguna condición política o personal.

En menor medida también en los objetivos “Trabajo decente y crecimiento económico” y “Ciudades y comunidades sostenibles”. Dado que el software libre ayuda a impulsar el crecimiento económico ya que pone en manos de los ciudadanos las herramientas para innovar y crear sus propios negocios al mismo tiempo que ayuda a tener un crecimiento

tecnológico basado en el bien común y, por lo tanto, más sostenible.

<b>Objetivos de Desarrollo Sostenibles</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>No Procede</b>
ODS 1. <b>Fin de la pobreza.</b>				<b>X</b>
ODS 2. <b>Hambre cero.</b>				<b>X</b>
ODS 3. <b>Salud y bienestar.</b>				<b>X</b>
ODS 4. <b>Educación de calidad.</b>			<b>X</b>	
ODS 5. <b>Igualdad de género.</b>				<b>X</b>
ODS 6. <b>Agua limpia y saneamiento.</b>				<b>X</b>
ODS 7. <b>Energía asequible y no contaminante.</b>				<b>X</b>
ODS 8. <b>Trabajo decente y crecimiento económico.</b>		<b>X</b>		
ODS 9. <b>Industria, innovación e infraestructuras.</b>	<b>X</b>			
ODS 10. <b>Reducción de las desigualdades.</b>	<b>X</b>			
ODS 11. <b>Ciudades y comunidades sostenibles.</b>		<b>X</b>		
ODS 12. <b>Producción y consumo responsables.</b>			<b>X</b>	
ODS 13. <b>Acción por el clima.</b>				<b>X</b>
ODS 14. <b>Vida submarina.</b>				<b>X</b>
ODS 15. <b>Vida de ecosistemas terrestres.</b>				<b>X</b>
ODS 16. <b>Paz, justicia e instituciones sólidas.</b>				<b>X</b>
ODS 17. <b>Alianzas para lograr objetivos.</b>				<b>X</b>

# Agradecimientos

Este trabajo no hubiera salido adelante sin dos personas a las que se lo quiero agradecer.

Gracias a Francisco Daniel Muñoz Escóí, mi tutor de TFG, por su paciencia, su dedicación y los conocimientos que ha compartido conmigo que han hecho posibles la realización de este trabajo.

Gracias especiales también a mi pareja, María Quijano Peralta. Sin su apoyo a lo largo del proceso de desarrollo de este estudio no me habría sido posible llevarlo hasta el final y cerrar este capítulo de mi vida años después de hacer el último examen del grado.