



YouTubeCrawlerTool: Aplicación web para habilitar el estudio del movimiento antivacuna en YouTube

Javier Sánchez Mendoza

Grado de ingeniería informática
Health IT

Carlos Luis Sánchez Bocanegra
José Antonio Morán Moreno

Junio de 2018



Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-SinObraDerivada
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Quiero dedicar este trabajo especialmente a:

Carolina

*Por empujarme a retomar mis estudios y, lo que es mas importante,
motivarme durante todo este tiempo.*

Amy, Luke y Jim

Por obligarme a salir a la calle de vez en cuando y estar siempre hay.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>YouTubeCrawlerTool: Aplicación web para habilitar el estudio del movimiento antivacuna en YouTube</i>
Nombre del autor:	<i>Javier Sánchez Mendoza</i>
Nombre del consultor/a:	<i>Carlos Luis Sánchez Bocanegra</i>
Nombre del PRA:	<i>José Antonio Morán Moreno</i>
Fecha de entrega:	<i>06/2018</i>
Titulación:	<i>Grado de ingeniería informática</i>
Área del Trabajo Final:	<i>Health IT</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave:	<i>antivacuna, crawler, YouTube.</i>
Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i>	
...	

Abstract (in English, 250 words or less):
...

Índice

1. Introducción	7
1.1. Contexto y justificación del Trabajo	7
1.2. Objetivos del Trabajo	8
1.3. Enfoque y método seguido	8
1.4. Planificación del Trabajo	9
1.5. Arquitectura tecnológica	12
1.6. Resumen de capítulos	12
2. Análisis y diseño	13
2.1. Metodología	13
2.2. Propuesta de la solución	13
2.2.1. Reuniones y entrevistas	13
2.2.2. Análisis de requisitos	13
2.2.3. Elección de YouTube como red social	13
2.2.4. La API de YouTube	13
2.2.5. Propuesta de interfaz de usuario	13
2.2.6. Visualización de vídeos en grafo	13
2.2.7. Casos de uso	13
2.3. Arquitectura de la solución	13
2.3.1. Arquitectura WEB	13
2.3.2. Elección de Neo4j como SGBD	13
2.3.3. Diseño capa de datos	13
2.3.4. Diseño capa de negocio	14
2.3.5. Diseño API RESTful	14
2.3.6. Diseño capa de presentación	14
3. Desarrollo	15
3.1. Entorno de desarrollo	15
3.2. Pruebas de concepto	15
3.2.1. POCTwitterCrawler	15
3.2.2. POCYouTubeCrawler	15
3.2.3. POCYouTubeCrawlerNeo4j	15
3.3. Estructura de la aplicación	15
3.4. Acceso a datos	15
3.5. Crawler de YouTube	15
3.6. Visualización de vídeos en grafo	15
3.7. Otras funcionalidades	16
3.8. Capa de seguridad	16
3.9. Capa de presentación	16
3.10. Definición y ejecución de pruebas	16

4. Implementación y puesta en funcionamiento	17
4.1. Manual de instalación y requerimientos	17
4.2. Servidor de explotación	17
4.3. Formación y sensibilización	17
4.4. Funcionalidades no implementadas	17
4.5. Propuesta de mejoras	17
5. Conclusiones	18
5.1. Conclusiones del trabajo	18
5.2. Grado de cumplimiento de los objetivos	18
5.3. Seguimiento de la planificación y metodología	18
5.4. Opinión del proyecto	18
6. Glosario	19
7. Bibliografía	20
8. Anexos	21

Índice de figuras

1.	Ejemplo del backlog del proyecto en Trello	9
2.	Listado de tareas y diagrama de Gantt	10
3.	Listado de tareas	10

1. Introducción

En esta sección se detalla el proyecto, la motivación de la elección de la temática escogida y la planificación y estructuración del mismo.

1.1. Contexto y justificación del Trabajo

Desde la introducción de la vacunación como método preventivo de enfermedades han existido entidades y grupos de personas que se han opuesto a ella y han dudado de su efectividad o propósito [1]. Hoy en día el activismo anti-vacunación (conocido también como movimiento antivacunas) ha vuelto a la actualidad y se encuentra en auge en algunas regiones tales como Europa o Estados Unidos, cobrándose en el peor de los escenarios víctimas mortales a causa de enfermedades que se creían erradicadas y que han vuelto a surgir [2][3].

Para hacer posible el estudio y comprensión de las motivaciones del movimiento antivacuna y luchar contra su desinformación, se propone el desarrollo de una aplicación que permita la recolección de grandes cantidades de datos de la actividad realizada por parte de este colectivo en redes sociales con el fin de hacer posible su posterior tratamiento y estudio para obtener valor añadido. Para tal fin, en este proyecto contamos con la colaboración de Johanna Milena Rodríguez Vera estudiante de master en Telemedicina que asume el rol de analista de datos (*data scientist* [4]) en el desarrollo de su trabajo final de máster titulado *Evaluación de la información sanitaria en vacunas disponible en las redes sociales - YouTube* y que actúa a la vez como cliente de la aplicación desarrollada en el presente proyecto.

Hoy en día las redes sociales han puesto al alcance de los analistas de datos una gran cantidad de información disponible para ser analizada, una de las problemáticas a las que se quiere hacer frente es la obtención de dichos datos de forma efectiva. Para ello se propone hacer uso de interfaces de programación de aplicaciones (abreviado como *API* [5] en inglés) ofrecidas públicamente por distintas redes sociales de tal forma que el proceso resulte transparente para el usuario final, permitiéndole la extracción a este problema.

La obtención de grandes volúmenes de datos nos lleva también a la problemática que surge en su almacenamiento en bases de datos tradicionales y su posterior procesamiento. Para habilitar al usuario final el correcto acceso a la información obtenida se estudian las ventajas que aporta el uso de bases de datos *NoSQL* [6] para este cometido, al ser diseñadas especialmente para manejar enormes cantidades de datos. Proyecto que se enmarca dentro de la problemática de la obtención, almacenamiento y procesamiento de grandes volúmenes de datos (*Big Data* [7]) y su posterior acceso.

1.2. Objetivos del Trabajo

El objetivo principal del proyecto es proporcionar una aplicación web que permita a la clienta obtener de forma usable y transparente la información que necesite de la red social *YouTube* para poder llevar a cabo el estudio de patrones de comportamiento entre las diferentes movimientos anti y pro vacuna.

Entre los objetivos secundarios del proyecto se encuentra la exploración de otras redes sociales y proporcionar una herramienta lo suficientemente genérica para que pueda ser utilizada en la investigación realizada por Johanna así como en otras investigaciones futuras de distinta temática.

Algunos objetivos concretos que se han querido lograr son los siguientes:

- Investigar que funcionalidades aportan las *API* públicas ofrecidas por *YouTube* y analizar como se pueden utilizar para la obtención de la información requerida.
- Determinar como almacenar y acceder de forma eficiente a la gran cantidad de información que se obtendrá.
- Permitir la recolección de información según criterios de búsqueda proporcionados por el usuario final.
- Habilitar la gestión, visualización y exportación de datos obtenidos en distintos procesos de extracción para su posterior análisis en herramientas especializadas.
- Ofrecer herramientas de visualización para el análisis y comprensión de los datos obtenidos.
- Proporcionar una interfaz de usuario usable que permita realizar las acciones requeridas por el usuario final.

1.3. Enfoque y método seguido

Para la realización del proyecto se ha seguido el marco ágil de desarrollo *scrum* [8]. Al adoptar esta metodología como marco de trabajo nos ha permitido, a diferencia de otras metodologías lineales de desarrollo como pueden ser los modelos en cascada, poder desarrollar el proyecto de forma flexible permitiéndonos adaptar la planificación inicial del proyecto en los casos necesarios para adecuarse a los nuevos requerimientos.

La forma en la cual se aplico la metodología *scrum* en el proyecto esta condicionada por los integrantes del equipo de desarrollo, en el cual la figura del *product owner*, *scrum master* y desarrollador recaen sobre la figura del

alumno que presenta el actual proyecto descrito (Javier Sánchez Mendoza), mientras que la figura del cliente estará representada por una analista de datos (Johanna Milena Rodríguez Vera) y el consultor de los mismos (Carlos Luis Sánchez Bocanegra) como *stakeholder*.

Siguiendo la metodología *scrum*, se realizaron iteraciones (comúnmente conocidos como *sprints*) de una semana de duración donde, en la finalización de los mismos, se realizaron reuniones online para revisar y aprobar las tareas realizadas (*sprint review*) y definir las tareas para la siguiente iteración (*sprint planning*). Para gestionar las tareas a realizar (*backlog*) se decidió utilizar la herramienta online *Trello* [9]:

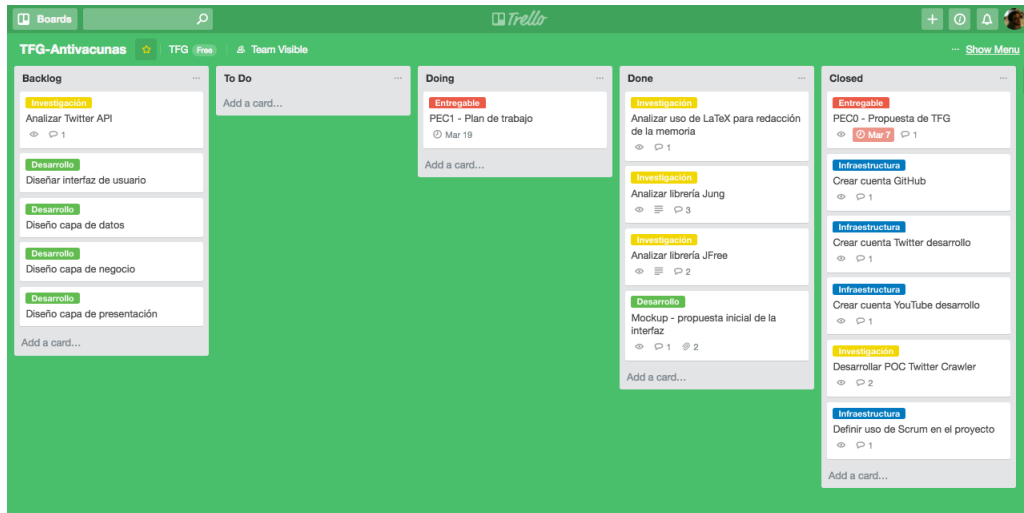


Figura 1: Ejemplo del backlog del proyecto en Trello

1.4. Planificación del Trabajo

En la realización del proyecto se propuso inicialmente seguir una planificación tentativa tal y como se detalla en el diagrama de *Grantt* facilitado en las figuras dos y tres:

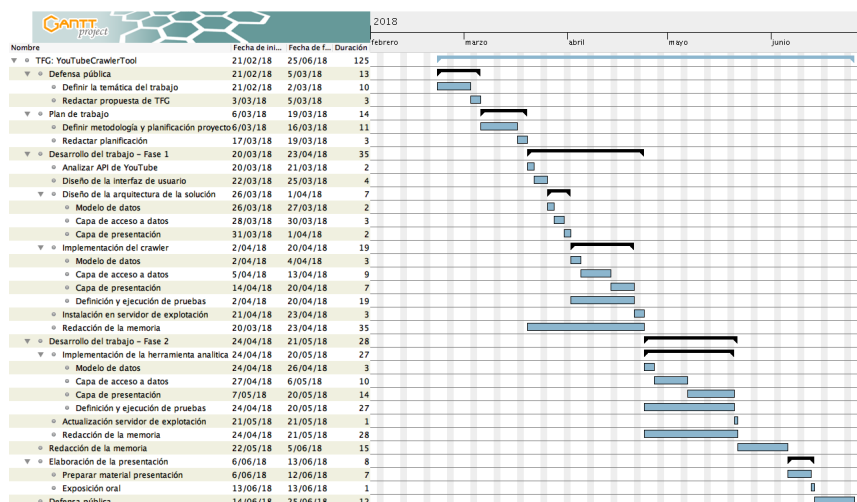


Figura 2: Listado de tareas y diagrama de Gantt

Nombre	Fecha de ini...	Fecha de f...	Duración
TFG: YouTubeCrawlerTool	21/02/18	25/06/18	125
Defensa pública	21/02/18	5/03/18	13
Definir la temática del trabajo	21/02/18	2/03/18	10
Redactar propuesta de TFG	3/03/18	5/03/18	3
Plan de trabajo	6/03/18	19/03/18	14
Definir metodología y planificación proyecto	6/03/18	16/03/18	11
Redactar planificación	17/03/18	19/03/18	3
Desarrollo del trabajo - Fase 1	20/03/18	23/04/18	35
Análisis API de YouTube	20/03/18	21/03/18	2
Diseño de la interfaz de usuario	22/03/18	25/03/18	4
Diseño de la arquitectura de la solución	26/03/18	1/04/18	7
Modelo de datos	26/03/18	27/03/18	2
Capa de acceso a datos	28/03/18	30/03/18	3
Capa de presentación	31/03/18	1/04/18	2
Implementación del crawler	2/04/18	20/04/18	19
Modelo de datos	2/04/18	4/04/18	3
Capa de acceso a datos	5/04/18	13/04/18	9
Capa de presentación	14/04/18	20/04/18	7
Definición y ejecución de pruebas	2/04/18	20/04/18	19
Instalación en servidor de explotación	21/04/18	23/04/18	3
Redacción de la memoria	20/03/18	23/04/18	35
Desarrollo del trabajo - Fase 2	24/04/18	21/05/18	28
Implementación de la herramienta analítica	24/04/18	20/05/18	27
Modelo de datos	24/04/18	26/04/18	3
Capa de acceso a datos	27/04/18	6/05/18	10
Capa de presentación	7/05/18	20/05/18	14
Definición y ejecución de pruebas	24/04/18	20/05/18	27
Actualización servidor de explotación	21/05/18	21/05/18	1
Redacción de la memoria	24/04/18	21/05/18	28
Redacción de la memoria	22/05/18	5/06/18	15
Elaboración de la presentación	6/06/18	13/06/18	8
Preparar material presentación	6/06/18	12/06/18	7
Exposición oral	13/06/18	13/06/18	1
Defensa pública	14/06/18	25/06/18	12

Figura 3: Listado de tareas

De la planificación inicial cabe destacar que la redacción de la memoria se diseñó como una tarea evolutiva que se desarrollaría durante todas las fases del proyecto pero mas intensamente durante la antepenúltima fase dedicada exclusivamente a su redacción. Además, durante las dos fases de desarrollo se planificaron dos tareas recurrentes para la definición y realización de pruebas de calidad del producto a realizar durante todo el ciclo de desarrollo.

Como se puede observar, en el diagrama facilitado en cada entrega se pretendían conseguir unos hitos concretos. La relación de los mas destacables por entrega son los siguientes:

- **Definición de los contenidos del trabajo:** Redacción propuesta TFG.
- **Plan de trabajo:** Redacción planificación.
- **Desarrollo del trabajo – Fase 1:** Instalación en servidor de explotación de la primera versión de la aplicación con funcionalidad de *crawler* implementada.
- **Desarrollo del trabajo – Fase 2:** Actualización en servidor de explotación de la versión final de la aplicación con funcionalidad analítica implementada.
- **Redacción de la memoria:** Entrega de la memoria del proyecto.
- **Elaboración de la presentación:** Realizar exposición oral.
- **Defensa pública:** Defender públicamente el proyecto.

Los riesgos detectados en la planificación inicial se concentraban principalmente en la consecución del hito definido en la primera fase de desarrollo. Para permitir a la clienta de la aplicación poder empezar a recopilar datos para su investigación lo antes posible, se decidió realizar la instalación de la aplicación desarrollada en un entorno de explotación en dos fases distintas, una con la funcionalidad del *crawler* y otra con la funcionalidad analítica implementada. La demora en la primera fase de desarrollo podría comprometer el éxito de la investigación del cliente. Para mitigar este riesgo se realizo el seguimiento del mismo durante las diferentes iteraciones en esa fase de desarrollo.

Cabe destacar que el diagrama proporcionado en esta sección representa la estimación inicial de la planificación del proyecto de forma tentativa y fue sujeto a modificaciones al inicio de cada nueva fase de desarrollo para garantizar el éxito del proyecto. En la sección de conclusiones del presente documento en el apartado 5.3 se realiza una comparación entre la planificación inicial y sus respectivas revisiones.

1.5. Arquitectura tecnológica

Para la consecución de los objetivos definidos se ha desarrollado una aplicación web bautizada como *YouTubeCrawlerTool* la cual ha sido diseñada por capas.

La capa de negocio de dicha aplicación ha sido desarrollada con tecnología Java EE [10] implementada mayormente utilizando proyectos del marco de desarrollo Spring Framework [11] entre otros. En esta capa se hace uso intensivo de servicios externos en forma de API pública ofrecida por YouTube con el fin de consumir dicha API para recolectar los vídeos requeridos por el usuario para su estudio.

En la capa de presentación se ha utilizado el lenguaje JavaScript [12] con un gran uso de JQuery [13] para modificar el DOM de las vistas y realizar llamadas asíncronas a la capa de negocio.

Finalmente, en la capa de datos se utiliza una base de datos orientada a grafos [14] la cual nos permite persistir los vídeos obtenidos en forma de grafo junto con sus relaciones además de otras informaciones derivadas y necesarias para el uso y funcionamiento la aplicación.

En los siguientes apartados se profundiza en la arquitectura y el diseño de la aplicación introducidos en esta sección.

1.6. Resumen de capítulos

En los próximos capítulos se detalla el trabajo realizado juntamente con los productos obtenidos y sus conclusiones.

La relación de capítulos es la siguiente:

- **Análisis y diseño:** Explicación de la metodología de diseño escogida así como los pasos que se llevaron a cabo para definir una propuesta a la clienta y su posterior análisis para acabar definiendo la arquitectura de la solución.
- **Desarrollo:** Descripción del desarrollo efectuado y de las decisiones realizadas durante el proceso.
- **Implementación y puesta en funcionamiento:** Detalle de los productos obtenidos con las instrucciones para su correcta puesta en funcionamiento además de las acciones de formación realizadas.
- **Conclusiones:** Sumario de resultados obtenidos y conclusiones sobre el trabajo realizado.

2. Análisis y diseño

En estos capítulos, hay que describir los aspectos más relevante del diseño y desarrollo del proyecto, así como de los productos obtenidos. La estructuración de los capítulos puede variar según el tipo de Trabajo.

En cada apartado es muy importante describir las alternativas posibles, los criterios utilizados para tomar decisiones y la decisión tomada.

En caso de que corresponda, se incluirá un apartado de “Valoración económica del trabajo”. Este apartado indicará los gastos asociados al desarrollo y mantenimiento del trabajo, así como los beneficios económicos obtenidos. Hacer un análisis final sobre la viabilidad del producto.

2.1. Metodología

//TODO: Describir la metodología elegida para el desarrollo del proyecto (Dirigida a usuario final??)

2.2. Propuesta de la solución

2.2.1. Reuniones y entrevistas

//TODO: Describir cada una de las reuniones y del proceso seguido

2.2.2. Análisis de requisitos

2.2.3. Elección de YouTube como red social

2.2.4. La API de YouTube

2.2.5. Propuesta de interfaz de usuario

<https://github.com/jsanchezmend/TFGAntivacunas/tree/master/Mockups/>
15:04

2.2.6. Visualización de vídeos en grafo

2.2.7. Casos de uso

2.3. Arquitectura de la solución

2.3.1. Arquitectura WEB

2.3.2. Elección de Neo4j como SGBD

2.3.3. Diseño capa de datos

<https://github.com/jsanchezmend/TFGAntivacunas/tree/master/Dise%C3%B1o/CapaDatos>

2.3.4. Diseño capa de negocio

<https://github.com/jsanchezmend/TFGAntivacunas/tree/master/Dise%C3%B1o/CapaNegocio>

2.3.5. Diseño API RESTful

2.3.6. Diseño capa de presentación

<https://github.com/jsanchezmend/TFGAntivacunas/tree/master/Dise%C3%B1o/CapaPresentacion>

3. Desarrollo

//TODO: Hablar de patrones usados (factory y etc) y frameworks utilizados

3.1. Entorno de desarrollo

Aplicación web (con POCYouTubeCrawlerNeo4j instalado): <http://youtubecrawlertoolwebapp.azurewebsites.net>

Servidor base de datos Neo4j: <http://51.136.48.142:7474/browser>
Usuario: neo4j Password: Y01t1b3cr4wl3rt00l Consulta de ejemplo: MATCH (n) RETURN n

3.2. Pruebas de concepto

- **Twitter crawler:** <https://github.com/jsanchezmend/TFGAntivacunas/tree/master/POCTwitterCrawler>
- **YouTube crawler:** <https://github.com/jsanchezmend/TFGAntivacunas/tree/master/POCYouTubeCrawler>
- **Visualización en grafo:** <https://github.com/jsanchezmend/TFGAntivacunas/tree/master/POCYouTubeCrawler>
- **Neo4j:** <https://github.com/jsanchezmend/TFGAntivacunas/tree/master/POCYouTubeCrawlerNeo4j>

3.2.1. POCTwitterCrawler

3.2.2. POCYouTubeCrawler

3.2.3. POCYouTubeCrawlerNeo4j

3.3. Estructura de la aplicación

3.4. Acceso a datos

// Spring data, entities, repositorios y POJOS

3.5. Crawler de YouTube

// Explicar conexión con YouTube y como se ha implementado el crawler

3.6. Visualización de vídeos en grafo

// Búsqueda de videos y su representación en grafo Cytripode.js

3.7. Otras funcionalidades

// Accose a videos, favoritos, canales, etc..

3.8. Capa de seguridad

//Spring security

3.9. Capa de presentación

//Plantillas JSP con Themleaf y requests Ajax con JQuery y jquery para modificar el dom

3.10. Definición y ejecución de pruebas

4. Implementación y puesta en funcionamiento

4.1. Manual de instalación y requerimientos

4.2. Servidor de explotación

4.3. Formación y sensibilización

4.4. Funcionalidades no implementadas

// Stats

4.5. Propuesta de mejoras

//API error handling, mejores logs, mas filtros de analisis y por base de datos, performance del grafo, UI en general, accesibilidad y usabilidad. inteligencia artificial para categorizar automaticamente los videos uilizando, por ejemplo, un modelo baso en reglas.

5. Conclusiones

Este capítulo tiene que incluir:

- Una descripción de las conclusiones del trabajo: Qué lecciones se han aprendido del trabajo?.
- Una reflexión crítica sobre el logro de los objetivos planteados inicialmente: Hemos logrado todos los objetivos? Si la respuesta es negativa, por qué motivo?
- Un análisis crítico del seguimiento de la planificación y metodología a lo largo del producto: Se ha seguido la planificación? La metodología prevista ha sido la adecuada? Ha habido que introducir cambios para garantizar el éxito del trabajo? Por qué?
- Las líneas de trabajo futuro que no se han podido explorar en este trabajo y han quedado pendientes.

5.1. Conclusiones del trabajo

// mencionar conclusiones de Johanna // Puntos fuertes y debiles // Criticas dificultades // ha merecido la pena etc...

5.2. Grado de cumplimiento de los objetivos

5.3. Seguimiento de la planificación y metodología

//TODO: Gant tentativo vs gant final (tomar prestado de los informes de seguimiento)

5.4. Opinión del proyecto

6. Glosario

Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

7. Bibliografía

Referencias

- [1] https://es.wikipedia.org/wiki/Controversia_de_las_vacunas (07/03/2018)
- [2] <http://www.elmundo.es/cataluna/2015/06/27/558e5fb2e2704ea41e8b4576.html> (07/03/2018)
- [3] <https://buenavibra.es/movida-sana/salud/italia-sarampion-movimientos-antivacunas> (16/03/2018)
- [4] https://es.wikipedia.org/wiki/Ciencia_de_datos (07/03/2018)
- [5] https://es.wikipedia.org/wiki/Interfaz_de_programacion_de_aplicaciones (07/03/2018)
- [6] <https://es.wikipedia.org/wiki/NoSQL> (07/03/2018)
- [7] <https://es.wikipedia.org/wiki/Macrodatos> (07/03/2018)
- [8] [https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)) (16/03/2018)
- [9] <https://trello.com> (16/03/2018)
- [10] <http://www.oracle.com/technetwork/java/index.html> (24/05/2018)
- [11] <https://spring.io/> (24/05/2018)
- [12] <https://es.wikipedia.org/wiki/JavaScript> (24/05/2018)
- [13] <https://jquery.com/> (24/05/2018)
- [14] https://es.wikipedia.org/wiki/Base_de_datos_orientada_a_grafos (24/05/2018)

8. Anexos

Listado de apartados que son demasiado extensos para incluir dentro de la memoria y tienen un carácter autocontenido (por ejemplo, manuales de usuario, manuales de instalación, etc.)

Dependiente del tipo de trabajo, es posible que no haya que añadir ningún anexo.