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

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:	*YouTubeCrawlerTool: Aplicación web para habilitar el estudio del movimiento antivacuna en YouTube*
Nombre del autor:	*Javier Sánchez Mendoza*
Nombre del consultor/a:	*Carlos Luis Sánchez Bocanegra*
Nombre del PRA:	*José Antonio Morán Moreno*
Fecha de entrega:	*06/2018*
Titulación:	*Grado de ingeniería informática*
Área del Trabajo Final:	*Health IT*
Idioma del trabajo:	*Español*
Palabras clave:	*antivacuna, crawler, YouTube.*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

...

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

Índice

Índice de figuras

1. Introducción

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 victimas 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 clienta 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 ingles) 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 estudían 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 decidio utilizar la herramienta online *Trello* [9]:

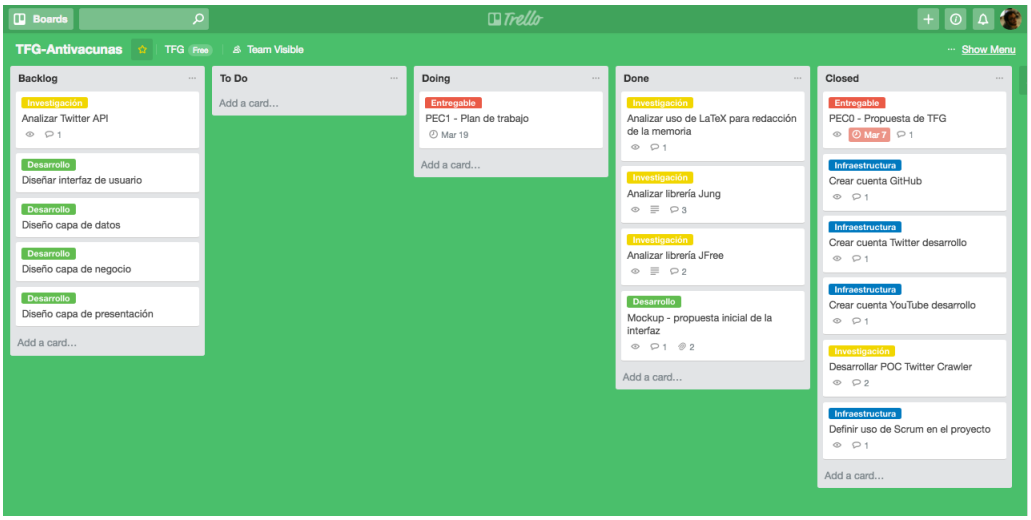
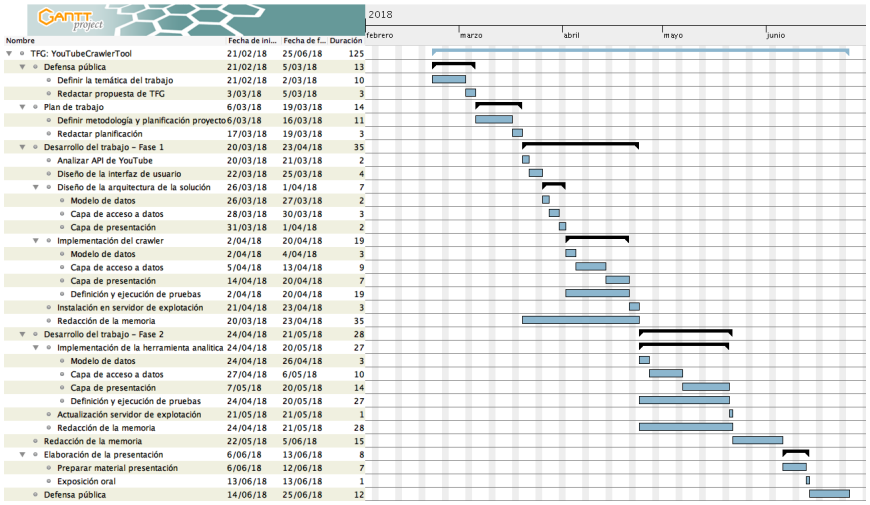


Figura 1: Ejemplo del backlog del proyecto en Trello

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:



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
Analizar 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 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
Analizar 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 diseño 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 API Rest habilitada en la capa de negocio para tales efectos.

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 ademas de otras informaciones derivadas y necesarias para el uso y funcionamiento de 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 ademas de las acciones de formación realizadas.
- **Conclusiones:** Sumario de resultados obtenidos y conclusiones sobre el trabajo realizado.

2. Análisis y diseño

2.1. Metodología

Para realizar el diseño de la aplicación se opto de entre diferentes posibilidades el enfoque definido por la filosofía del 'Diseño centrado en el usuario' [15].

El diseño centrado en el usuario, como bien indica su nombre, se caracteriza por conocer a fondo a los futuros usuarios de una aplicación para diseñar un producto que resuelva sus necesidades y expectativas buscando en todo momento conseguir la mayor satisfacción del usuario posible. Se trata de un proceso iterativo y cíclico por fases en donde en cada una de ellas se utilizan distintas técnicas para conseguir los objetivos propuestos.

En nuestro proyecto se ha decidido seguir esta y no otra metodología de diseño tales como el 'Diseño centrado en la actividad' o el 'Diseño centrado en el uso' debido a que, como en nuestro caso la clienta de la aplicación va a ser también la usuaria final de la misma, se ha decidido realizar el análisis de la aplicación centrada en ella y sus necesidades por encima de la actividad que se llevara a cabo o el uso.

La relación de fases y técnicas utilizadas ha sido la siguiente:

- **Definir contexto de uso:** El objetivo de esta fase es el de determinar que necesidades pretende la usuaria final que la aplicación resuelva y a que se va a destinar su uso. La técnica escogida para la realización de esta fase fueron las entrevistas con la usuaria final que se realizaron mediante videoconferencia sobretodo durante las diferentes *sprint reviews* al termino de cada *sprint*.
- **Especificar requisitos:** En la siguiente fase se definen los requisitos del sistema a partir de la información recogida en la fase previa. Para los requisitos funcionales se opto por recogerlos como 'casos de uso' [16].
- **Diseñar el producto:** En esta fase se diseñan y implementan los requisitos definidos en la fase anterior ya sea con el objetivo de proporcionar una solución final o una propuesta de solución a ser refinada en sucesivas iteraciones. Las técnicas utilizadas en esta fase fueron la creación de maquetas para evaluar la solución y el prototipado mediante pruebas de concepto realizadas con el objetivo de estudiar posibles soluciones antes de realizar su implementación en la aplicación.

- **Poner a prueba el producto:** Finalmente, en esta fase se pone a prueba el producto obtenido. Para hacerlo, se definieron y ejecutaron pruebas de integración diseñadas teniendo en cuenta los casos de uso definidos previamente y se realizaron test con usuarios para evaluar su grado de satisfacción.

Gracias al enfoque escogido fue posible encontrar respuestas a preguntas sobre las expectativas que la usuaria tenia depositadas sobre la aplicación y que fueron de gran ayuda a la hora de diseñar la solución final. Algunas de las principales preguntas fueron:
¿Quiénes son los usuarios de la aplicación?
¿Cuáles son las tareas a realizar?
¿Qué funcionalidades se necesitan?
¿Qué información se necesita?

También cabe destacar que gracias a que tanto *scrum* como el diseño centrado en el usuario son dos procesos iterativos, resulto fácil integrar esta metodología dentro del marco ágil de desarrollo.

2.2. Propuesta de la solución

2.2.1. Reuniones y entrevistas

Tal y como se ha introducido en la sección 2.1 sobre la metodología de diseño utilizada, para definir el contexto de uso y conocer las necesidades a ser cubiertas por la aplicación, se realizaron varias entrevistas con la clienta y los *stakeholders* donde la gran mayoría de ellas fueron dentro del contexto de *scrum* como reuniones de *sprint review* y *sprint planning*.

A continuación se resumen las entrevistas y reuniones realizadas junto con los principales temas tratados y decisiones tomadas:

Fecha: 05/03/2018
Hora de inicio: 21:00
Hora final: 21:45
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta
- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Que conocimiento sobre el movimiento antivacunas se quiere obtener.

- Que se pretende hacer con la información recolectada.
- Definición de los objetivos de la aplicación.

Decisiones:

- La clienta definirá los criterios de búsqueda y el conocimiento que debe ser obtenido por la aplicación.
- El análisis de los datos los realizara la clienta con herramientas especializadas.
- El objetivo principal de la aplicación es la obtención de datos mediante búsquedas en redes sociales.
- La aplicación debe ofrecer funcionalidad para exportar los datos recolectados a otras herramientas.

Fecha: 12/03/2018
Hora de inicio: 21:00
Hora final: 21:30
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta
- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Alcance del estudio a realizar.
- Sobre que redes sociales debe centrarse el estudio.

Decisiones:

- Se decide adoptar el marco de trabajo *scrum* para la realización del proyecto.
- El estudio se centrara inicialmente en una sola red social a determinar.
- En la aplicación sera posible ejecutar varios procesos de recolección de datos al mismo tiempo.
- Se incorporara una sección para analizar los datos obtenidos de forma visual (por definir).

- El *product owner* realizara una propuesta inicial de la interfaz de usuario.

Fecha: 19/03/2018
Hora de inicio: 21:00
Hora final: 21:30
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta

Temas tratados:

- Comentarios sobre la propuesta inicial de la interfaz de usuario realizada por el *product owner*.
- Pros y contras sobre la elección de *Twitter* como red social para el estudio.

Decisiones:

- Elección de *Twitter* como red social a utilizar en el estudio.
- La aplicación incluirá una visualización en forma de grafo para poder analizar visualmente las relaciones existente en la información recolectada y descubrir patrones.
- El *product owner* debe estudiar la viabilidad de utilizar *Twitter* para la consecución de los objetivos.

Fecha: 26/03/2018
Hora de inicio: 21:00
Hora final: 21:45
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta

Temas tratados:

- Debido a las limitaciones de uso detectadas en la API de *Twitter*, se propone utilizar la red social *YouTube* como alternativa.

- Definición del grafo a utilizar y que elementos actuaran como nodo y aristas.

Decisiones:

- Elección de *YouTube* como red social a utilizar en el estudio.
- El *product owner* debe estudiar la viabilidad de utilizar *YouTube* para la consecución de los objetivos.
- Actualizar la propuesta de interfaz de usuario para reflejar el cambio de red social.

Fecha: 05/04/2018
Hora de inicio: 21:00
Hora final: 21:45
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta
- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Comentarios sobre la propuesta inicial de la interfaz de usuario realizada por el *product owner*.
- Analizar criterios de búsqueda y información devuelta por la API de *YouTube*
- Detección de vídeos duplicados.
- Detectar usuarios *influencers* a partir de la información obtenida.
- Avances en la definición de los componentes del grafo.

Decisiones:

- Criterios de búsqueda de *YouTube* a utilizar para recolectar los vídeos.
- Campos a almacenar de cada vídeo.
- Incorporar funcionalidad para ver resumen de las búsquedas realizadas junto con la información recolectada con pre visualización de vídeos.
- Utilizar una variable pre calculada (bautizada como "*scopeRange*") para determinar el tamaño de los nodos al ser visualizados en el grafo.

Fecha: 12/04/2018
Hora de inicio: 21:30
Hora final: 22:15
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta
- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Comentarios sobre la visualización en grafo.
- Identificación de vídeos pro y anti vacunación.

Decisiones:

- Queda definida la visualización en grafo.
- Los contenidos a visualizar en el grafo podrán ser filtrados.
- Se define funcionalidad para etiquetar los vídeos recolectados utilizando categorías previamente definidas por el usuario en la aplicación.
- En la aplicación habrá dos tipos de usuarios, usuarios anónimos que no podrán realizar acciones de escritura ni borrado y usuarios registrados que podrán realizar todas la acciones.
- Incorporar visualizaciones estadísticas sobre el uso de las categorías.
- Realización de una ultima propuesta de interfaz de usuario que recoja los últimos cambios propuestos junto con la visualización en grafo.

Fecha: 19/04/2018
Hora de inicio: 21:30
Hora final: 22:00
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta
- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Comentarios sobre la propuesta de interfaz de usuario.
- Priorización de funcionalidades.

Decisiones:

- La clienta da por aprobada la propuesta de interfaz de usuario.
- Definición de valores por defecto al realizar las búsquedas de contenidos.
- Las funcionalidades estadísticas y de visualización de canales quedan asignadas con una prioridad secundaria en relación a otras funcionalidades.
- Se va a buscar la colaboración de un estadista para definir la formula de la variable "*scopeRange*".

Fecha: 26/04/2018
Hora de inicio: 21:00
Hora final: 21:30
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta
- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Seguimiento de la implementación de la aplicación
- Seguimiento de la definición de la fórmula para calcular el alcance de los vídeos ("*scopeRange*").

Decisiones:

Fecha: 03/05/2018
Hora de inicio: 21:00
Hora final: 21:30
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta

- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Seguimiento de la implementación de la aplicación.
- Seguimiento de la definición de la fórmula para calcular el alcance de los vídeos ("*scopeRange*").

Decisiones:

Fecha: 10/05/2018
Hora de inicio: 20:30
Hora final: 21:00
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta
- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Seguimiento de la implementación de la aplicación.
- Seguimiento de la definición de la fórmula para calcular el alcance de los vídeos ("*scopeRange*").

Decisiones:

- Instalación de la aplicación en entorno de explotación durante la próxima semana.
- Mientras no se disponga de la formula para la variable "*scopeRange*", utilizar formula alternativa definida por la clienta.

Fecha: 17/05/2018
Hora de inicio: 20:30
Hora final: 21:30
Asistentes:

- Carlos Luis Sánchez Bocanegra: *stakeholder*
- Johanna Milena Rodríguez Vera: clienta

- Javier Sánchez Mendoza: *product owner*, *scrum master* y desarrollador

Temas tratados:

- Presentación de la aplicación desarrollada y formación a la clienta.

Decisiones:

- Se decide utilizar la formula alternativa propuesta por la clienta para la variable "*scopeRange*".
- Añadir nueva funcionalidad de favoritos.
- Realización de cambios en la visualización de los listados de los vídeos.

En los siguientes apartados se detallan algunas de las decisiones tomadas en estas reuniones las cuales tuvieron mas repercusión en el resultado final de la aplicación.

2.2.2. Elección de YouTube como red social

A la hora de escoger una red social para realizar el estudio del movimiento antivacuna nos encontramos con una gran variedad de opciones, tales como *Facebook*, *Twitter* o *YouTube* para nombrar solo algunas.

Johanna, clienta de la aplicación y la encargada de realizar el estudio, mostró su predilección inicial por *Twitter*. En su criterio, *Twitter* presentaba una estructuración de la información que favorecía su posterior análisis y estudio al basarse, principalmente, un mensaje (*tweet*) de los campos titulo y descripción, hecho que facilitaba la categorización de los contenidos obtenidos. Por otro lado, el uso extensivo de etiquetas en esta red social (*hashtags*) simplificaría el proceso de búsqueda de contenidos. Por lo referente al alcance de la red social, Johanna consideraba que a nivel de usuarios y actividad relacionada con la vacunación, *Twitter* superaba a otras redes sociales tales como *YouTube*.

Debido a estas consideraciones, *Twitter* fue escogida inicialmente como la red social a utilizar en el proyecto. Por este motivo, inicialmente la aplicación desarrollada se llamaba *TwitterCrawlerTool* y las primeras versiones de la propuesta de interfaz de usuario se habían realizado teniendo en consideración a *Twitter* como red social escogida. Incluso, una prueva de concepto llego a desarrollarse: POCTwitterCrawler.

Con la elección realizada y teniendo desarrollada una prueba de concepto, se procedió al estudio en profundidad de la API de *Twitter* y fue entonces

cuando se descubrieron limitaciones sobre su uso. Concretamente, *Twitter* define tres niveles de uso: *Standard*, *Premium* y *Enterprise*; de todas ellas solo la opción *Standard* es completamente gratuita pero, en este caso, con unas severas restricciones de uso a la hora de buscar mensajes [17]. Algunas de las restricciones son:

- Máximo de 100 *"tweets"* por búsqueda.
- Información disponible solo de los últimos 7 días.
- Solo permite búsqueda textual, no se permite búsqueda temporal.

A causa de estas restricciones, resultaba imposible poder recabar información con una antigüedad inferior a siete días o realizar comparaciones en el tiempo sobre la evolución de los movimientos pro y anti vacunas, hecho que es requerido en el estudio. Ante los descubrimientos realizados se decidió explorar otras alternativas al uso de *Twitter*.

Fue entonces que la opción de utilizar *YouTube* como red social de estudio en el proyecto se considero en profundidad. Y es que aunque se llegara anteriormente a la conclusión de que *Twitter* tenia un alcance mayor en numero de usuarios y potencial de contenidos a obtener, no se debe desestimar tampoco el alcance de *YouTube* que, si bien inicialmente no era considerada como una red social al uso, a día de hoy el numero de usuarios que no solo visualizan sus vídeos sino que también comparten contenidos y comentarios crece día a día, convirtiendo a *YouTube* como una buena opción para encontrar la información requerida. Y en lo referente a la estructuración de la información, debido a que juntamente con los vídeos se provee un titulo y una descripción no era necesario cambiar el enfoque dado inicialmente en este sentido.

Por lo referente a la API ofrecida por *YouTube* para la obtención de contenidos, los criterios de búsqueda disponibles son mucho mas amplios que los ofrecidos por *Twitter*, permitiéndonos entre otras posibilidades, filtrar los contenidos a obtener por rangos de fechas. No existe limitación a la hora de obtener contenidos independientemente de su fecha de publicación (en todo caso posterior a 14/02/2005, fecha de fundación de *YouTube*). Y en relación a limitaciones en el volumen de información a obtener, *YouTube* no impone limitaciones por búsqueda, lo que posibilita obtener toda la información requerida sobre un termino en concreto. En su defecto, *YouTube* utiliza un sistema de cuotas que se aplica a periodos de tiempo en concreto [18], por ejemplo, en su versión gratuita permite realizar hasta 1.000.000 de operaciones de lectura por día que, tal y como se ha demostrado, han sido suficientes para el uso dado a la aplicación desarrollada. Para estudiar su viabilidad en el proyecto, se realizo una prueba de concepto con resultados

satisfactorios: POCYouTubeCrawler.

En el apartado 3.2 se detalla en profundidad los servicios de la API de *YouTube* utilizados y el modo en que la aplicación los consume para obtener contenidos.

2.2.3. Visualización de vídeos en grafo

Otra de las decisiones de diseño mas debatidas fue la definición de una vista en la aplicación que permitiera de forma visual analizar como se relacionan los movimientos anti y pro vacuna en la red social de *YouTube*.

La motivación principal de la misma es la de proporcionar una funcionalidad con la cual fuera posible estudiar las vías de acceso a los contenidos y observar que movimiento tiene mas alcance de audiencia en *YouTube*. Para tales efectos, se llego a la conclusión que una visualización en formato de grafo donde fuera posible distinguir los dos movimientos a estudio seria la forma mas efectiva de representar dicha información.

Para difinir el grafo, necesitábamos analizar que componentes actuarían como nodo y que representarían las aristas entre ellos, ademas de tomar otras decisiones como si se incorporarían pesos al grafo y si este seria dirigido o no. Para ayudar en la definición del grafo, se realizaron propuestas que se apoyaron con pruebas realizadas manualmente por parte de la clienta con contenidos reales:

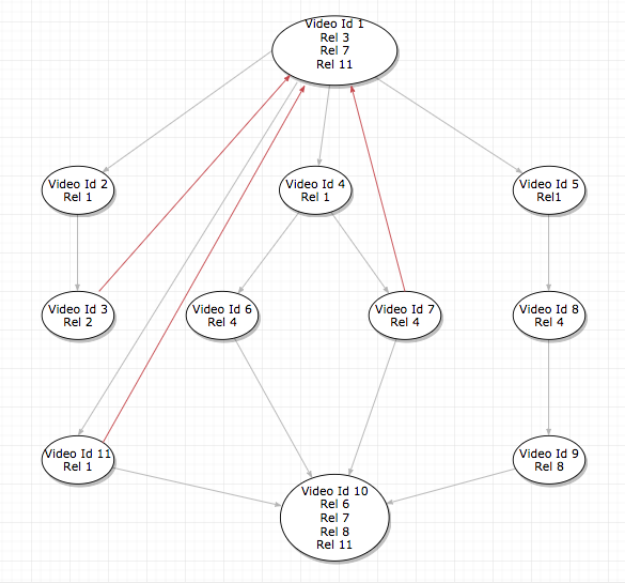


Figura 4: Propuesta grafo dirigido

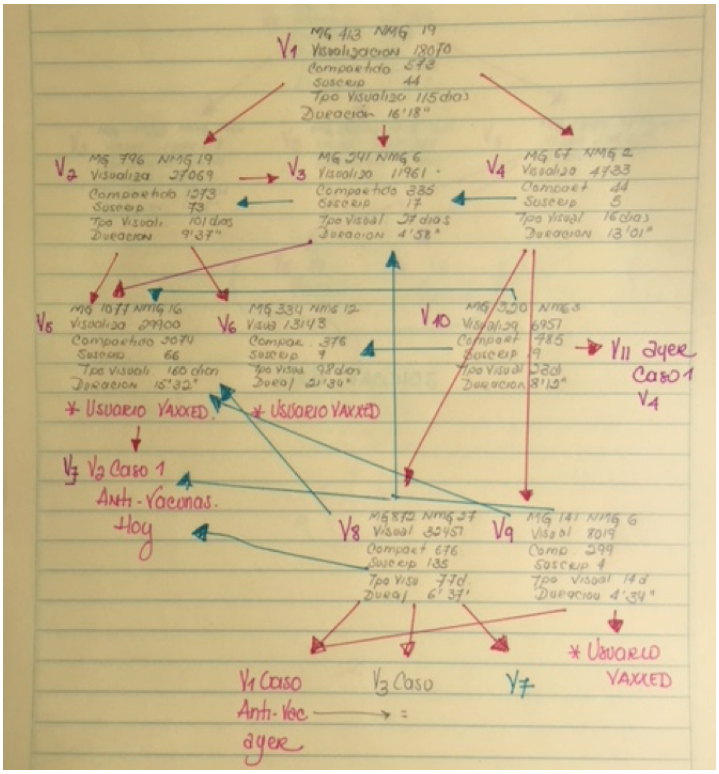


Figura 5: Maqueta de grafo dirigido

Como conclusiones del análisis realizado se definió el grafo a representar con las siguientes características:

- **Tipo:** Grafo dirigido sin pesos.
- **Nodos:** Vídeos y canales (canales de forma opcional).
- **Aristas:** Entre dos vídeos representa que desde el vídeo origen es posible acceder al vídeo destino (mediante la recomendación realizada por *YouTube* como vídeos relacionados). Y entre un vídeo y un canal representa que el vídeo (nodo origen) ha sido publicado por el canal relacionado (nodo destino).
- **Visualización:** En nodos tipo vídeo, su tamaño sera determinado por su alcance de audiencia y el color de representación sera determinado por su categoría (anti o pro vacuna). En el caso de nodos tipo canal su tamaño y representación no sera determinado por ninguna de sus características, por lo que todos los canales se visualizaran con el mismo formato pero diferenciados de los nodos tipo vídeo. Sera posible ver una pre visualización del contenido al hacer clic en el.

Para estudiar su viabilidad, se realizo una prueba de concepto con resultado favorable: POCYouTubeCrawler.

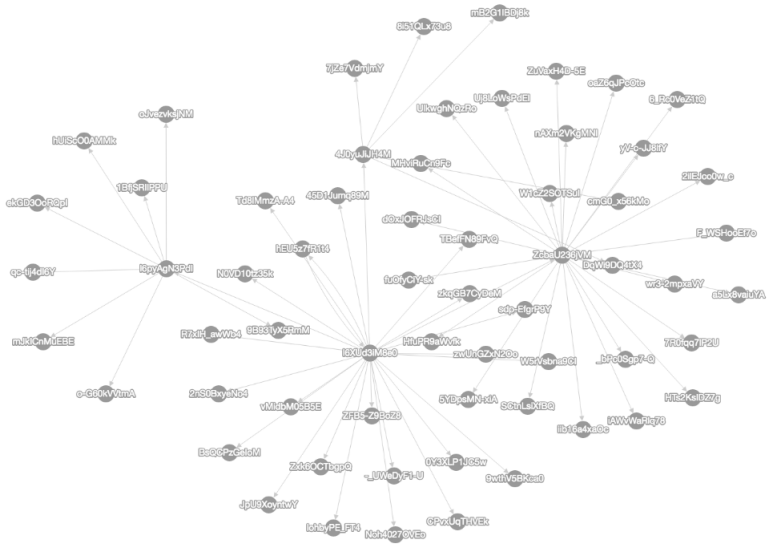


Figura 6: Prueba de concepto de grafo dirigido

En la propuesta de grafo detallada, se detectaron dos requerimientos encubiertos necesarios para su realización: la necesidad de categorizar vídeos y la definición de una variable que nos permitiera determinar el alcance de audiencia de un vídeo para definir su tamaño en el grafo.

Para la categorización de los vídeos, se diseño una funcionalidad genérica que permitiera al usuario definir las categorías necesarias con las cuales posteriormente poder etiquetar los vídeos. El inconveniente de esta solución es que para poder visualizar correctamente el grafo y identificar los grupos de nodos, se requiere realizar previamente una tarea de categorización manual de los vídeos. Después de analizar la problemática, debido a que el volumen de datos requeridos para realizar el estudio se determino en una cifra menor a mil, se acepto la solución como viable pero, en este caso, se debía tener presente esta circunstancia en la planificación del proyecto.

Por otro lado, para definir el tamaño de visualización del nodo se estudio la creación de una variable apodada como *scopeRange*. Dicha variable, debía representar dentro de un rango de valores valido, el alcance o popularidad obtenido por un vídeo en concreto. Para ello se estudio poder identificar a los usuarios mas influyentes de *YouTube* (conocidos como *influencers*), pero finalmente se decidió utilizar la información estadística ofrecida por *YouTube*

para cada vídeo, la cual se compone de los campos:

- ***viewCount*:** Numero de visualizaciones del vídeo.
- ***likeCount*:** Numero de personas a las cuales le ha gustado el vídeo.
- ***dislikeCount*:** Numero de personas a las cuales no le ha gustado el vídeo.
- ***commentCount*:** Numero de comentario que ha recibido el vídeo.

Para la definición de la formula se pidió la colaboración de un estadista. Pero debido a que la fecha final de desarrollo del proyecto se aproximaba y aun no se disponía de la colaboración, se opto por aplicar una formula definida por la clienta que en las pruebas realizadas demostró efectividad:

$$scopeRange = \frac{likeCount}{dislikeCount}$$

Con un valor mínimo definido de 10 para asegurar la visualización del vídeo en el grafo.

A modo ilustrativo, a continuación se adjunta una imagen de la implementación del grafo en la aplicación:

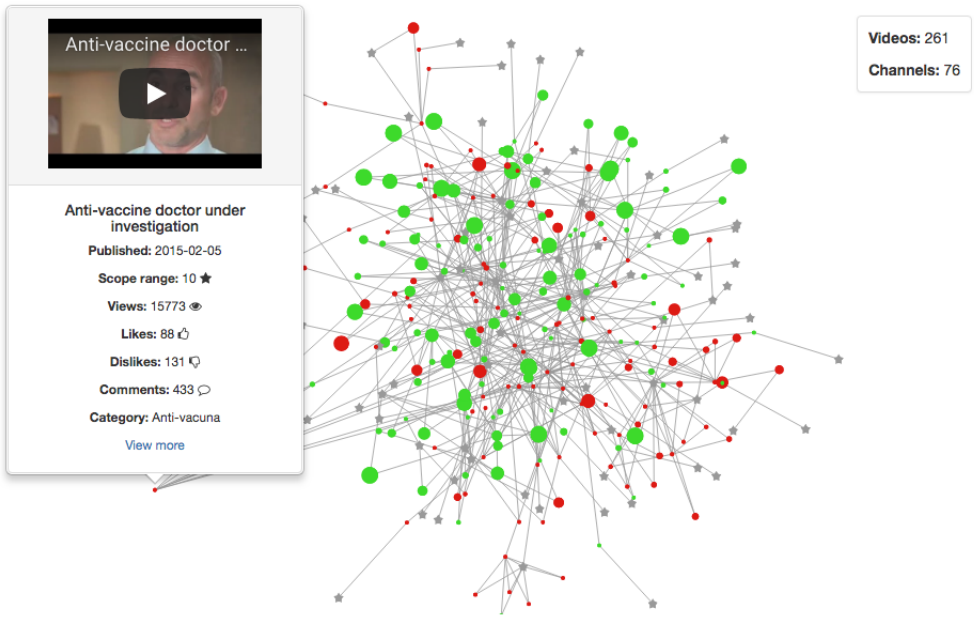


Figura 7: Implementación del grafo en la aplicación

2.2.4. Casos de uso

2.2.5. Propuesta de interfaz de usuario

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

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 framworks 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. Análisis YouTube Data API

3.3. 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

3.3.1. POCTwitterCrawler

3.3.2. POCYouTubeCrawler

3.3.3. POCYouTubeCrawlerNeo4j

3.4. Estructura de la aplicación

3.5. Acceso a datos

// Spring data, entities, repositorios y POJOS

3.6. Crawler de YouTube

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

3.7. Visualización de vídeos en grafo

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

3.8. Otras funcionalidades

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

3.9. Capa de seguridad

//Spring security

3.10. Capa de presentación

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

3.11. 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) (16/03/2018)

[9] https://trello.com (16/03/2018)

[10] http://www.oracle.com/technetwork/java/index.html (23/05/2018)

[11] https://spring.io/ (23/05/2018)

[12] https://es.wikipedia.org/wiki/JavaScript (23/05/2018)

[13] https://jquery.com/ (23/05/2018)

[14] https://es.wikipedia.org/wiki/Base_de_datos_orientada_a_grafos (23/05/2018)

[15] https://es.wikipedia.org/wiki/Dise%C3%B1o_centrado_en_el_usuario (24/05/2018)

[16] https://es.wikipedia.org/wiki/Caso_de_uso (24/05/2018)

[17] https://developer.twitter.com/en/docs/tweets/search/overview (26/05/2018)

[18] https://developers.google.com/youtube/v3/getting-started?hl=es-419#quota (26/05/2018)

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.