HERRAMIENTA PARA VISUALIZAR LA EVOLUCIÓN DE PRODUCCIÓN DE CONOCIMIENTO ON-LINE EN WIKIS

Trabajo de fin de grado del Grado en Ingeniería Informática
Facultad de Informática, Universidad Complutense de Madrid
Curso 2016-2017

MARCEL R. SANDOVAL ROSALES

Directores del trabajo: Javier Arroyo Gallardo

Samer Hassan Collado

ÍNDICE

Resumen....4
Abstract....5
1. Introducción....6
1.1. Contexto....6
1.2. Motivación....7
1.3. Objetivos....8
1.4. Estructura del documento....8
2. Introduction....10
2.1. Context....10
2.2. Motivations....11
2.3. Objectives....11
2.4. Document structure....12
3. Estado del arte....12
3.1. Producción colaborativa....12
3.2. Comunidades colaborativas....13
3.3. Wikia....16
3.4. Herramientas de análisis de comunidades colaborativas....17
3.4.1. Bitergia Cauldron....17
3.4.2. Edgesense....18
3.4.3. Wiki activity monitor (WAM)....19
3.4.4. WikiDAT....19
3.4.5. ChartsUp....20
4. Metodologías del proyecto....21
4.1. Organización del trabajo....21
4.2. Desarrollo iterativo incremental....21
4.3. Visualización de datos....22
4.4. Software libre....23
5. Tecnologías....24
5.1. Python....24
5.1.1. NumPy....24
5.1.2. SQLite3....25
5.1.3. Bokeh....25

5.2. HTML 26
5.3. CSS 26
5.4. Sublime Text 3 27
5.5. Github 27
6. La fuente de datos y su preprocesamiento 28
6.1. La fuente de los datos 28
6.2. Estructura del data-dump 29
6.3. Preprocesamiento de los datos 30
7. Arquitectura de la aplicación 32
7.1. Parseo de los datos 32
7.2. Almacenamiento y acceso a los datos 34
7.3. Creación de las gráficas 35
7.3.1. Gráficas 36
7.3.2. Widgets 36
7.3.3. Interacciones 38
7.4. Arquitectura de servidor de Bokeh 39
8. Interfaz de la aplicación 40
8.1. Evolution 41
8.2. Statistics 46
9. ejemplo de uso 50
10. Conclusiones y trabajo futuro 52
10.1. Conclusiones 52
10.2. Trabajo futuro 52
10. Conclusions 54
10.1. Conclusions 54
10.2. Future work 54
10. Bibliography 56
Anexo: Manual de instalación 59
Prerrequisitos de software 59
Ejecución del código 59

ÍNDICE DE FIGURAS

Figura 1. Consumo colaborativo 13
Figura 2. Distribución de los usuarios 15
Figura 3. Ejemplo de dashboard de Bitergia....18
Figura 4. Interfaz gráfica de edgesense....18
Figura 5. Top 5 wikis según puntuación WAM 19
Figura 6. ChartsUp....20
Figura 7. Ejemplo de visualización con Bokeh 25
Figura 8. Estructura de la plantilla HTML para Bokeh 26
Figura 9. Ejemplo de sección de descarga de data-dump 28
Figura 10. Estructura del data-dump 29
Figura 11. Ejemplo de etiqueta <siteinfo> (vista en Fig. 10) rellena....29
Figura 12. Ejemplo de página con una única revisión 30
Figura 13.Ejemplo de revisión. 30
Figura 14. Esquema de las relaciones entre los distintos elementos de la aplicación..32
Figura 15. Ejemplo de consulta SQL 35
Figura 16. Ejemplo de creación de gráfica....36
Figura 17. Creación del Slider 37
Figura 18. Creación del CheckBoxGroup 37
Figura 19. Asignación del gestor de eventos al Slider 38
Figura 20. Gestor de eventos del Slider 38
Figura 21. Asignación de los gestores de evento a los CheckBox 38
Figura 22. Gestor de evento de los CheckBox 39
Figura 23. Arquitectura de servidor Bokeh 39
Figura 24. Ejemplo de pestaña de evolución 42
Figura 25. Ejemplo de gráfica acumulada 43
Figura 26. Ejemplo de gráfica acumulada 44
Figura 27. Panel de páginas. 44
Figura 28. Panel de ediciones 45
Figura 29.Panel de usuarios....46
Figura 30. Panel de tamaño 46
Figura 31. Ejemplo de pestaña de estadísticas 47
Figura 32. Top Usuarios 49
Figura 33. Distribución de las ediciones entre los usuarios....49
Figura 34. Top Páginas....49
Figura 35. Distribución de las ediciones entre las páginas 49
Figura 36. Evolución de páginas totales de Laguna Negra wiki 50
Figura 37. Evolución del número total de ediciones de Lguna Negra wiki....50
Figura 38. Evolución del promedio de ediciones por página de Laguna Negra wiki....50
Figura 39. Nuevos usuarios registrados de Laguna Negra wiki 51
Figura 40. Nuevos usuarios anónimos de Laguna Negra wiki 51
Figura 41. Usuarios registrados vs usuarios anónimos de Laguna Negra wiki 51

RESUMEN

Las comunidades colaborativas forman cada vez una parte más importante de nuestra realidad. Bien sea a través de software libre (GitHub) o por plataformas de conocimiento compartido (Wikipedia, StackOverflow) lo cierto es que hoy en día parece inevitable nutrirse de los contenidos generados por estas comunidades. Sin embargo, el conocimiento que tenemos del funcionamiento organizacional interno todavía se encuentra en un estado poco avanzado.

Existen multitud de estudios sobre Wikipedia, quizá el ejemplo más grande y de mayor impacto de estas comunidades. Sin embargo, no se debe caer en la tentación de extrapolar los hallazgos al resto de comunidades, menos populares y de contenido más específico. ¿Por qué algunas comunidades triunfan mientras otras se quedan en el olvido? ¿De qué depende el éxito de un proyecto colaborativo? ¿Cómo está relacionada la continuidad de los mismos con la actividad de sus usuarios? Éstas son sólo algunas de las preguntas que pueden surgir a la hora de analizar la evolución de distintas wikis.

Este proyecto pretende proporcionar una herramienta para el análisis un conjunto de comunidades tan grande como lo es Wikia (ahora conocida como FANDOM, se trata de una plataforma de wikis temáticas) que permita responder a algunas de las cuestiones planteadas y a cualquier otra que le pueda surgir al administrador de una comunidad. Además, también servirá para monitorizar el estado de la misma a lo largo del tiempo.

El resultado final es una aplicación de escritorio que se ejecuta en modo de servidor local y se despliega en un navegador web. Esta aplicación, desarrollada en Python, permite visualizar los datos obtenidos de la exportación de la base de datos de una wiki (conocida como data-dump), para así poder observar tanto la evolución de la comunidad a lo largo del tiempo de una forma global, como el estado de la misma en un momento concreto del tiempo.

Palabras clave: análisis de datos, minería de datos, Wikia, data dump, visualización, comunidades colaborativas, producción colaborativa, wiki, dashboard, Python

ABSTRACT

Collaborative communities are increasingly present in our daily life. Werther on open source software (GitHub) or in peer produced knowledge (Wikipedia, StackOverflow), it is inevitable to be a consumer of the contents generated by this communities. However, their organization is yet to be understood.

There are many studies about perhaps the greatest and most influential of them all: Wikipedia. However, one should not be tempted to assume that the findings about Wikipedia can are applicable to other, more specific, communities. Why do some of them triumph while most of them are abandoned? What determines the success of such communities? How can user activity determine the survival of the community? These are some of the questions that arise when analyzing this phenomenon.

This project intends to provide a tool for the analysis of a community such as those in Wikia (known as FANDOM nowadays) which could lead to answer some of the questions stated above as well as many others that may arise. Also, it will be especially useful to wiki administrators to monitor their ecosystem and analyze its evolution.

The final result is a desktop app executed as a local server that deploys on a web browser. This application, developed in Python, allows the user to visualize the data obtained from a wiki database export, also known as data-dump, so that he or she can observe the global evolution of the community as well as its status at a certain point in time.

Keywords: data analysis, Wikia, data mining, data dump, visualization, collaborative communities, peer production, wiki, dashboard, Python

1. INTRODUCCIÓN

1.1. Contexto

La era de la información ha supuesto un gran avance en la difusión del conocimiento, pero lo que mucha gente ignora es que también la producción se ha visto revolucionada. La producción colaborativa se ha visto potenciada gracias a Internet. Ésta es aquella en la que dos o más personas trabajan en el desarrollo de un proyecto, siguiendo una estructura autoorganizada. En la actualidad son múltiples los ejemplos de producción colaborativa (software libre, Wikipedia…). Las comunidades colaborativas no se rigen por una jerarquía estricta que siguen las organizaciones más tradicionales. En cambio, su estructura más horizontal fomenta la participación e inclusión de los colaboradores.

En aquellos casos en los que el bien producido es digital, el modelo de producción resulta más propenso a atraer un mayor número de colaboradores. En particular, en el área de Informática el ejemplo más claro es el del software libre. Los proyectos desarrollados como software libre se prestan a que todo aquel que considere que tiene algo que aportar tenga la oportunidad de hacerlo. Así, gracias al alcance masivo que proporciona Internet, cientos o incluso miles de personas aportan su granito de arena, lo que en conjunto supone un avance considerable y eficaz de los proyectos.

Existen multitud de proyectos de software libre, algunos de la talla de GNU/Linux [1] o WordPress (el 27% de las páginas en la web usan Wordpress), lo que puede llegar a dar una idea del alcance de la producción colaborativa. Pero más allá de la importancia tecnológica que tiene el software libre, la transparencia es un aspecto clave. Plataformas de interés social como Decide Madrid [2] garantizan la legitimidad del proceso gracias a su condición de software libre.

Este modelo de producción va mucho más allá del desarrollo de software o el área de la informática en general. La producción colaborativa de conocimiento ha supuesto una nueva etapa en la humanidad a la hora de compartir y distribuir conocimiento. Aquellas páginas web que pueden ser editadas de forma fácil por los usuarios, muchas veces incluso de forma anónima, conocidas como wikis [3] han tenido un papel clave en el auge esta era.

Aunque se pueden encontrar infinidad de plataformas, cabe destacar la mundialmente conocida Wikipedia [4]. Wikipedia es una enciclopedia online que utiliza el software de MediaWiki [5] y que genera su contenido de forma colaborativa. Fue creada en 2001 por Jimmy Wales [6] y Larry Sanger [7] y a día de hoy cuenta con más de 37 millones de artículos en más de 280 idiomas distintos [8].

Lo impresionante es que este gran cúmulo de conocimiento ha sido generado por millones de usuarios sin conseguir nada a cambio, y resulta útil a un número de personas mucho más elevado. Una de las características principales de la producción colaborativa, en la que se incluye Wikipedia, es que la cantidad de colaboradores es ínfima comparada con el volumen de consumidores. Además, es una comunidad autorregulada, en la que son los propios usuarios los que garantizan la integridad de la información.

Además de Wikipedia, el fenómeno de las wikis ha sido muy extenso y existen múltiples plataformas basadas en este concepto. En general, estas plataformas suelen

dedicarse a profundizar en temas sobre los que Wikipedia sólo aporta una noción general. El ámbito que abarcan es de lo más variopinto, desde el mundo de la educación, como por ejemplo Wikiversity [9], el mundo sanitario (Ganfyd [10]) o aspectos menos genéricos como es el caso de las llamadas wikis locales, que contienen información sobre una región geográfica concreta.

En particular, este proyecto se centra en el estudio de Wikia, creada también por Jimmy Wales, en 2006. Ésta recoge múltiples comunidades del fenómeno fandom (término que se refiere a la subcultura de fans de un determinado tema).

El fenómeno de la producción colaborativa supone una revolución socioeconómica de los modelos ya conocidos y asentados. Por esta razón, ha sido objeto de estudio, sobre todo en el aspecto de la organización interna dado que, en un principio, no cuenta con regulaciones rígidas que la estructuren. Una de las grandes ramas de la producción colaborativa es el software libre [11] [12], pero en este proyecto nos centramos en la producción colaborativa de conocimiento, en particular en las wikis. Aunque existen estudios sobre otras wikis [13], el foco de los estudios ha sido hasta el momento Wikipedia, la gran enciclopedia de Internet.

En definitiva, la producción colaborativa es un fenómeno muy actual por lo que todavía no se tiene un conocimiento demasiado avanzado. Así, herramientas como las aquí presentadas resultan fundamentales para conseguir profundizar en el entendimiento de las comunidades colaborativas basándose en datos cuantitativos.

1.2. Motivación

Como acabamos de ver, la producción colaborativa de conocimiento está en constante evolución, por lo que es un tema de gran interés para su investigación. La gran mayoría de estudios que se han realizado se han centrado en Wikipedia [14], sin embargo, otras grandes comunidades no han sido analizadas en detalle.

La creación de esta herramienta se hace necesaria para aquellos que quieran comparar si el modelo que sigue la gran enciclopedia de Internet es extrapolable a otros núcleos de información con un abanico menos extenso, como es el caso de Wikia. También puede resultar interesante a administradores de las wikis o usuarios activos que simplemente quieran estar al tanto del estado de su comunidad, o incluso a desarrolladores de MediaWiki (tecnología que utilizan tanto Wikia como Wikipedia) como un punto de partida para un plugin de monitorización.

A día de hoy, no existe ninguna herramienta que permita visualizar la evolución de datos como el número de usuarios o ediciones de forma que se puedan relacionar y quizá comprender mejor el progreso de la producción de conocimiento. Así, la herramienta presentada en este proyecto no sólo cumple una función de por sí, sino que cubre una necesidad existente en el contexto en el que se encuentra.

Por otro lado, y ya más centrados en Wikia, los datos que nos proporcionan al descargarnos una base de datos (conocido como data-dump) resultan difícilmente interpretables, a pesar de ser bastante inteligibles. Por tanto, el trabajo de este proyecto intenta cubrir otro agujero del mundo de las wikis: una herramienta que permita procesar y simplificar estos complejos data-dumps para poder analizarlos de forma más sencilla.

1.3. Objetivos

Todas las necesidades definidas anteriormente han sido cubiertas esta aplicación. Para ello, se definieron una serie de objetivos con el fin de tener fijar una meta antes de establecer el camino a seguir.

El principal objetivo del trabajo es proporcionar una interfaz gráfica que sirva a los analistas de las comunidades de Wikia en la búsqueda de comprensión de la estructura y evolución de las mismas. Esto se puede desgranar en los siguientes objetivos:

- **Creación de una herramienta** de procesado de los data-dumps proporcionados con Wikia para que resulten más inteligibles.
- **Procesado y análisis** de los datos para obtener métricas e información más reveladora de la que proporciona la observación de los datos en bruto.
- Plasmar una **visualización** de los datos que resulte relevante a simple vista, además de proporcionar al usuario la posibilidad de interactuar con algunos de los elementos visuales, para que la herramienta permita explorar la evolución de una comunidad y así entender los factores de éxito o fracaso de la misma, además del estado en el que se encuentra en cualquier momento desde su nacimiento.

1.4. Estructura del documento

En este documento se pretende explicar con detalle el desarrollo del proyecto, para lo que se habla sobre el contexto en el que se encuentra, así como aspectos más técnicos del mismo. Para ello, se ha estructurado de la siguiente manera:

1. **Introducción:** en este capítulo se introduce al lector en el contexto en el que se desarrolla el proyecto, así como la dirección que sigue y por qué.
2. **Estado del arte**: se explicará más en detalle el contexto, así como otras herramientas de ámbito similar a la desarrollada en este proyecto.
3. **Métodos**: consistirá en una explicación de las distintas metodologías que han tomado parte en este proyecto.
4. **Tecnologías**: en esta sección se explicarán las principales tecnologías utilizadas para el desarrollo de la herramienta.
5. **Procesamiento de los datos**: es la base de la herramienta. Se explicará la estructura de los datos que trata la misma, así como el procesado que se realiza para su posterior análisis.
6. **Arquitectura de la aplicación**: en este apartado se explicará el diseño de la aplicación a nivel técnico, explicando los obstáculos encontrados y las decisiones tomadas para seguir adelante.
7. **Manual de instalación**: se explicará el proceso a seguir para desplegar la aplicación.
8. **Interfaz de usuario**: constará de un recorrido por la aplicación en sí, definiendo la utilidad de cada elemento visual. Contará además con un

ejemplo de caso de uso con una wiki concreta para poder mostrar el alcance que puede tener la herramienta.

9. **Conclusiones**: para culminar se hará un sumario de los resultados y se dedicará una sección a comentar el trabajo futuro, i.e. algunos posibles caminos por los que se podría continuar el trabajo.

2. INTRODUCTION

2.1. Context

The information era has brought with it a great advance in the diffusion of knowledge, but what many people ignore is that there has also been a revolution in the production system. Peer production, that is when two or more people work together towards a common goal without a structural organization, has been rising thanks to the internet. There are lots of day to day examples, such as Wikipedia or any open source software project. Collaborative communities do not have a strict governance policy as traditional institutions do. This more horizontal hierarchy gets more people to contribute, since they do it on their own terms.

Particularly, when talking about peer production of digital content, this model tends to grow substantially. In computer science, peer production is more a reality than in any other field. Free and open source projects are open to anyone who wish to contribute, and thanks to the Internet this transforms into thousands of people contributing. These contributions, however small they may be, means that the project moves forward considerably faster than traditional projects.

There are many, well known, open source projects, some of them as big as GNU/Linux or WordPress (around 27% of all web pages uses this content manager). Moreover, open source software has great implications regarding transparency. Social platforms such as Decide Madrid, can be monitored by regular citizens thanks to the fact that the source code is available to anyone who wishes to see it.

This production model goes far beyond software development or computer science in general. Peer production of knowledge has started a new era in human history when it comes to sharing and distributing knowledge. Those websites easily editable by users, sometimes even anonymously, known as wikis have had a key role in the rise of this era.

Although there are multiple platforms for knowledge sharing, the most noticeable of all is Wikipedia. This encyclopedia uses MediaWiki software and its content is generated through peer production. It was created in 2001 by Jimmy Wales and Larry Sanger and it has more than 37 million articles in more than 280 languages till this day.

The fact that all this knowledge has been produce voluntarily, makes this phenomenon worth studying. One of the main aspects of peer production, which includes Wikipedia, is the great difference between the number of contributors and the number of consumers. What's more, this community is autoregulated: the integrity of the information is guaranteed by the users.

As Wikipedia emerged, countless other wiki platforms did as well. In general, these platforms tend to go deeper into subjects that Wikipedia only gives a glance. The scope can be almost anything, from academics, like Wikiversity, to medical wikis, such as Ganfyd, or less generic subjects like in local wikis (also known as city wikis), which gather information about a local region.

In particular, this project focuses on the study of Wikia, also founded by Jimmy Wales, in 2006. This platform has over 300.000 communities within about fandom, a term used to describe the subculture of fans of a certain subject.

Peer production carries a socioeconomic revolution from the well-known and well-established models. For that, it has been the subject of countless studies, especially on the organizational aspect, because it has no strict rules that regulates its structure but even so it does have one. One of the great examples of peer production is free and open source software, but this project focuses on the production of knowledge, wikis in particular. Although there are indeed studies about other wikis, the main focus of research has been Wikipedia, the great encyclopedia of the Internet.

In summary, peer production is a current phenomenon, which is why the understanding of it is not too profound. Therefore, tools like the ones presented in this project are vital to go deeper into these communities using quantitative data.

2.2. Motivations

As we just saw, peer production is in constant evolution, which makes it a great topic for research. Most of them, as previously stated, focus on Wikipedia, leaving other big communities out.

The development of this tool is a necessity for those who wish to explore if this other communities share some aspects with the great encyclopedia. Also, wiki administrators as well as other interested users can use this tool to monitor the activity of their community and have a closer look of its evolution. Even MediaWiki developers can use this as inspiration for a monitoring dashboard module.

Nowadays, there isn't a tool that fulfills this need of visualizing core data of the evolution of a wiki, such as number of users or of editions. Therefore, the presented tool not only is useful per se, but it was required by the context it is built in.

Also, the database hardly readable by the user, which implies another necessity: a tool which can process this data-dumps and make them readable for the average user.

After looking thoroughly, it is clear that the tool presented by this project needed to exist in order to better understanding the evolution of Wikia communities, and in order to compare them with Wikipedia.

2.3. Objectives

The main goal of this project is to provide a graphic interface which online community analysts can use as a tool to dig further on this subject. This can be accomplished thanks to the following objectives:

- The **creation of a tool** to transform Wikia data-dumps and make them understandable.

- To process and analyze the data to obtain more revealing metrics and information than what we can obtain directly from the data.
- Provide a **Data visualization** which is relevant at plain sight, as well as giving the user the possibility to interact with the visual element, thus allowing them to choose the data they see fit to analyze.

2.4. Document structure

This documents intents to give a detailed explanation of the development of this project. In order to do that it is necessary to speak about the context in which the tool is develop, as well as the technical details. The document has been structure with the following chapters:

1. **Introduction**: this is a brief introduction to the project, talking about its objectives and motivations.
2. **State of the art**: this chapter will be about the context of the project as well as similar tools.
3. **Methods**: the methodology followed in the course of the project.
4. **Technologies**: the technologies used in the development of the tool.
5. **Data processing:** description of the data used for the visualization.
6. **Architecture:** this chapter is about the design decisions and structure of the tool itself.
7. **Installation guide:** these are the steps to be followed in order to deploy the application.
8. **User interface:** in this chapter, the user interface of the tool is explained in detail.
9. **Conclusions:** summary of the result as well as future work.

3. ESTADO DEL ARTE

En este capítulo se describirá en detalle el contexto en el que se engloba el proyecto, es decir el de las comunidades colaborativas y la investigación en torno a ellas. Además, se hablará de algunas herramientas similares bien en utilidad o bien en diseño que han sido tenidas en cuentas a la hora de desarrollar la aplicación aquí presentada.

3.1. Producción colaborativa

La producción colaborativa, o peer production, es aquella que se da cuando dos o más personas participan en un proyecto en común. Hoy en día, gracias a las facilidades que nos ofrece Internet, los proyectos de producción colaborativa están cada vez más presentes. Múltiples y diversas organizaciones están utilizando este modelo para conseguir sus objetivos de una forma que, no sólo hace que el progreso sea mucho más rápido, sino que también supone hacerlo de un modo más democrático.

El término *peer production* fue acuñado inicialmente por el profesor Yockai Benkler, profesor de la Universidad de Harvard, y lo desarrolla en su libro La Riqueza de las Redes [15]. En este libro, recoge distintas características que describen este fenómeno socioeconómico emergente, potenciado gracias a las redes de comunicación que se han formado en la era de la comunicación.

Benkler defiende la necesidad de que estos proyectos sean abiertos para que su alcance sea máximo, por lo que muchos de ellos tienen licencias muy permisivas. Ésta característica implica una gran difusión, lo que significa que entra en contacto con más usuarios en potencia. Otro aspecto a tener en cuenta es que la asignación de tareas no es por asignación, si no que los contribuyentes trabajan en aquellas tareas en las que ellos mismos piensen que puedan aportar algo.

Vivimos en un momento ahora mismo en el que las comunidades no paran de crecer. Un claro ejemplo es el consumo colaborativo. Este se basa en la utilización temporal de los bienes más en lugar de su adquisición en propiedad. Así, plataformas como Uber [16] o Airbnb [17] son líderes en su sector y están poniendo en jaque a las instituciones más tradicionales.

Figura 1. Consumo colaborativo

3.2. Comunidades colaborativas

Entre los contribuyentes de un proyecto de producción colaborativa surge una comunidad. Esta comunidad, en un principio, no tiene definidas unas reglas que determinen su estructura, lo que da pie a una organización más horizontal que las tradicionales. Sin embargo, a medida que la comunidad evoluciona, parece inevitable que se formen núcleos entre los usuarios más activos. Un estudio de Kolbitsch y Maurer sobre las comunidades online [18] pone una cota al número de usuarios que

puede tener una comunidad antes de necesitar regulaciones: a partir de 150 personas las comunidades se hacen menos individualistas y tienden a la organización.

Uno de los casos más relevantes de producción colaborativa es el caso del software libre. Una de las principales comunidades de software libre es la de GitHub, un repositorio online que pone a la mano de todo el mundo el código de un determinado proyecto. Éste modelo de producción de software permite un avance mucho más eficaz y una transparencia que el software propietario no ofrece. Son muchas las motivaciones que pueden llevar a los usuarios a colaborar, desde la mejora del software para uso propio a la participación en proyectos de interés social.

Sin embargo, la producción colaborativa va mucho más allá del desarrollo software. La producción colaborativa de conocimiento se ha convertido en parte fundamental del día a día de las personas, bien sea como productores o colaboradores, o en la mayoría de los casos, como consumidores. Este conocimiento en su mayoría ha sido generado a través de wikis, por lo que es imprescindible hablar de la principal plataforma de este tipo: Wikipedia.

Wikipedia ejemplifica a la perfección el comportamiento de una gran comunidad colaborativa y, por tanto, ha sido objeto de múltiples estudios. Uno de ellos fue el realizado por Felipe Ortega [14], en el que analizaba el comportamiento de los usuarios en las principales Wikipedias del mundo. Como resultado observó una gran desigualdad en la contribución de los distintos usuarios a largo plazo. Lo que resulta especialmente interesante es que esta desigualdad no está relacionada en la mayoría de los casos con otros factores como el tamaño de la wiki o el número de contribuyentes. Esto implica que esta desigualdad es, en cierta medida inevitable.

Esta desigualdad no se limita únicamente a Wikipedia, sino a los proyectos colaborativos on-line en general. La gran mayoría de las aportaciones se concentra en un 1% de los usuarios, otro 9% de los usuarios aportan ocasionalmente, pero la gran mayoría de ellos se dedican únicamente a consumir el contenido generado y no a aportar ellos mismos. Es lo que se denomina la **regla del 90-9-1.**

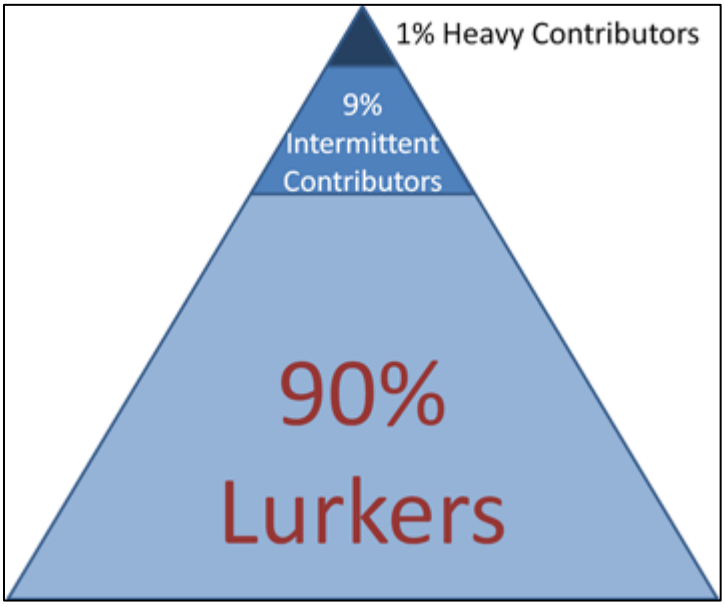


Figura 2. Distribución de los usuarios

Nielsen explica muy bien en un artículo [19] las principales consecuencias de esta desigualdad. El hecho de que la gran mayoría de contenidos de la web hayan sido creados por una minoría tiene unas implicaciones que pueden resultar muy impactantes. Resultados de búsquedas de internet, opiniones online o comentarios de clientes están presentes en el día a día del usuario medio de Internet. Sin embargo, si tenemos en cuenta la desigualdad de participación, la información que recibimos está representando únicamente a 1 de cada 100 usuarios.

Propone también algunos mecanismos para mitigar este reparto tan desigual. Es fundamental que al usuario medio le resulte lo más fácil posible contribuir, que no le exija un esfuerzo que ponga a prueba realmente sus ganas de participar. Incluso, si es posible, que participe de forma inconsciente, como por ejemplo la sección “la gente que compró esto también compró esto otro” de Amazon. Otra posible solución sería la recompensa por participación, pero ésta puede ser peligrosa si es demasiado atractiva porque puede generar avaricia por parte de los usuarios.

Como ya se ha mencionado, Wikipedia es quizá la comunidad colaborativa más grande que existe en la actualidad, con más de 4 millones de usuarios. Con semejante tamaño, no debe resultar sorprendente que haya surgido de forma natural una organización medianamente jerárquica. Sin embargo, esta estructura oligárquica no es ni mucho menos rígida, puesto que los propios administradores son cuestionados en todo momento.

Retomando el tema de la organización en las comunidades colaborativas, es interesante tener en cuenta a Jo Freeman [20] cuanto afirma que es imposible que no exista una estructura dentro de un grupo de personas, sin importar la razón que las une, su personalidad o cualquier otro factor.

Siguiendo esta línea, el trabajo de Mako Hill [13] nos proporciona un primer acercamiento al estudio de esta estructura en las comunidades de Wikia. Usando una muestra de 683 wikis, intenta demostrar si se cumple la ley de hierro de la oligarquía. Ésta fue desarrollada por el sociólogo alemán Robert Michels en su libro Political Parties [21] y asegura que a medida que una comunidad democrática crece, inevitablemente tiende hacia la oligarquía.

Las conclusiones del estudio de Mako Hill es que, en efecto, observando la desigualdad existente tanto en la participación como en el liderazgo de las wikis, llevan a que la comunidad tenga una estructura relativamente oligárquica. Para comprender bien las implicaciones que esto conlleva, es necesario conocer más en detalle el funcionamiento de la comunidad.

3.3. Wikia

Wikia es un servicio gratuito de alojamiento de páginas web construidas en forma de wikis, es decir, páginas fácilmente editables por cualquiera a través de un navegador web. En particular, Wikia aloja wikis temáticas sobre asuntos populares como pueden ser videojuegos o películas. Actualmente Wikia utiliza como nombre FANDOM (término que se refiere a la subcultura de aficionados de un determinado tema). Éste cambio se hizo para potenciar la marca y conseguir un volumen de usuarios mayor. Considerado la web más importante de fans, cuenta con más de 360.000 comunidades y más de 190 millones de visitas mensuales. Aunque las comunidades pueden ser de tamaños muy variados y temáticas muy diversas, no debe parecer descabellado buscar un comportamiento similar en todas ellas.

Fue fundada en 2004 por Jimmy Wales, fundador de Wikipedia, y Angela Beesly con el nombre de Wikicities, pero su nombre fue cambiado a Wikia en 2006. A lo largo de su historia, Wikia ha incorporado múltiples wikis independientes. Como ya se ha mencionado, sus wikis tienen contenido sobre la industria del entretenimiento, permitiéndose entrar en mucho más detalle que Wikipedia en general.

La edición en esta plataforma, como en cualquier wiki, se realiza a través de su interfaz web. No es necesario estar registrado para contribuir, aunque existe la posibilidad de que un administrador de la comunidad sí que imponga esta restricción. Aunque se puede ver justificado por el vandalismo, esta restricción afecta considerablemente a la producción colaborativa, habiendo casos en los que incluso implica el estancamiento de la wiki.

La comunidad de Wikia distingue distintos tipos de usuarios, que se definen por los privilegios que éstos tengan. Aunque existe la posibilidad de hacer tipos de usuario personalizados, a continuación, se explicarán los usuarios comunes:

- **Usuarios registrados:** son los usuarios de carácter general, que se han creado una cuenta. Tienen derecho a realizar acciones comunes como personalizar la apariencia de la wiki (para ellos), subir imágenes, tener un perfil, entre otras.

- **Usuarios autoconfirmados:** aquellos usuarios que lleven más de 4 días registrados se engloban en esta categoría, y se les permite editar páginas semi-protegidas, además de que se le deja de solicitar que rellenen captchas.
- Wikia mantiene una categoría de **usuarios de poder** para usuarios especialmente activos, pero que no tiene ninguna implicación de privilegios. Es una clasificación para uso interno de Wikia.
- **Administradores:** usuarios de confianza, generalmente elegidos democráticamente por la comunidad (pero designados finalmente por los burócratas), con privilegios importantes que incluyen repartir roles, editar el aspecto general de la wiki entre otras cosas. Los fundadores son automáticamente administradores. En caso de wikis abandonadas se puede conseguir mediante una solicitud.
- **Burócratas:** la principal utilidad de este rol es gestionar los distintos privilegios de los usuarios, aunque incluye los privilegios de otros roles.

Estos son sólo algunos ejemplos de tipo de usuario de los muchos que hay. Otros por ejemplo son los moderadores (de contenido, de discusiones…), los fundadores, los bots… En nuestro proyecto no se distinguen los tipos de usuario (salvo entre registrado y anónimo) puesto que solo tenemos información de la actividad del mismo, pero no de sus privilegios.

3.4. Herramientas de análisis de comunidades colaborativas

En este apartado se explicarán algunas herramientas que pueden resultar similares en algunos aspectos al resultado que se pretende obtener de este proyecto. Bien para buscar inspiración, como para tener claro qué es lo que ya existe sobre el tema, esta investigación previa era necesaria para desarrollar una herramienta que no sólo sea útil si no que aporte algo innovador al sector.

3.4.1. Bitergia Cauldron

Bitergia [22] es una compañía dedicada a la creación de soluciones para el análisis de datos de proyectos de desarrollo software. Su principal ámbito suelen ser proyectos de software libre, pero es también aplicable a empresas que quieran monitorizar el desarrollo de sus proyectos. Fue fundada por profesionales del grupo GSyC/Libresoft de la Universidad Rey Juan Carlos.

Figura 3. Ejemplo de dashboard de Bitergia

En particular, puede resultar más acorde con nuestros intereses su solución Bitergia Cauldron. Desarrollada como software libre, esta herramienta genera un dashboard con el que analizar la actividad de los proyectos de GitHub, como pueden ser los commits o las solicitudes de pull. La información se muestra gráficamente de forma clara y comprensible para el usuario, para facilitar la comprensión de la misma y que pueda sacarle el máximo partido.

3.4.2. Edgesense

Edgesense [23] es un módulo del gestor de contenidos Drupal [24] que permite visualizar gráficamente algunos datos sobre la actividad en los foros y comunidades de los sitios web. Desarrollado como software libre por Wikitalia [25], esta herramienta establece una red de conexiones entre los distintos participantes, lo que sirve para identificar los núcleos de actividad, distinguir conversaciones banales, entre otras cosas.

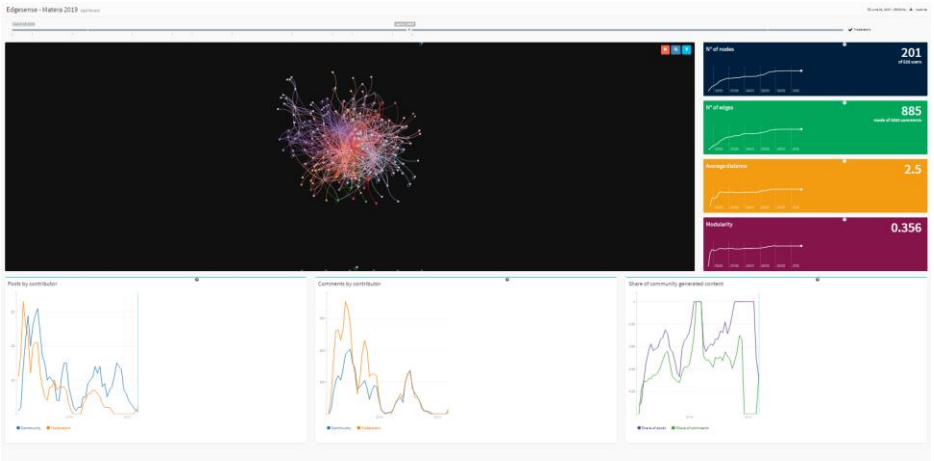
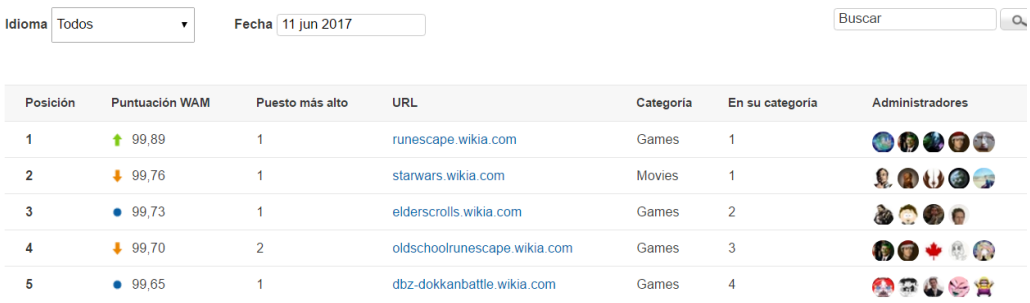


Figura 4. Interfaz gráfica de edgesense

Ya hablando más de los elementos visuales, además de la gráfica principal que sería la representación de la red, también muestra otras gráficas de evolución temporal. Además, tiene una barra deslizante en el tiempo para poder ver el estado de las interacciones en un determinado momento. Esta idea ha sido utilizada para este proyecto porque se ha considerado interesante poder visualizar no sólo la evolución de la comunidad, sino cómo se encontraba en determinado instante.

3.4.3. Wiki activity monitor (WAM)

Wikia proporciona una herramienta sencilla para comparar sus distintas comunidades. Ésta consiste simplemente en un ranking en función de lo que ellos llaman puntuación WAM. Aunque no especifican exactamente cómo se realiza el cálculo del mismo para evitar que las comunidades intenten manipular su puntuación. Sin embargo, lo que sí explican es que se calcula en función del tráfico, la actividad y el crecimiento de la wiki.



Posición	Puntuación WAM	Puesto más alto	URL	Categoría	En su categoría	Administradores
1	99,89	1	runescape.wikia.com	Games	1	
2	99,76	1	starwars.wikia.com	Movies	1	
3	99,73	1	elderscrolls.wikia.com	Games	2	
4	99,70	2	oldschoolrunescape.wikia.com	Games	3	
5	99,65	1	dbz-dokkanbattle.wikia.com	Games	4	

Figura 5. Top 5 wikis según puntuación WAM

Para nuestros intereses este ranking aporta mucho valor, puesto que nos dice directamente las wikis consideradas mejores en los aspectos ya mencionados. Utilizando la herramienta que se presenta en este proyecto, se pueden analizar esas wikis para investigar si existen patrones comunes en las wikis que triunfan.

3.4.4. WikiDAT

Wikipedia Data Analysis Toolkit es una herramienta desarrollada por Felipe Ortega, profesor de la Universidad Rey Juan Carlos. Se trata de una aplicación extensible para R y Python que permite analizar distintas métricas de una comunidad tan grande como es Wikipedia. Precisamente utilizando esta herramienta, Ortega pretende analizar factores que determinen el ascenso o abandono de proyectos online, en particular de Wikipedia [14].

La intención de este proyecto es proporcionar una herramienta de similar utilidad aplicada a las comunidades de FANDOM (Wikia) y poder comparar el comportamiento con el de otro tipo de proyectos como es la propia Wikipedia.

3.4.5. ChartsUp

ChartsUp [26] es el resultado del trabajo de fin de grado equivalente al aquí presentado, pero del curso 2015-2016. La idea detrás del mismo era similar: proporcionar un interfaz gráfico para el análisis de las wikis de Wikia. Sin embargo, el enfoque es distinto ya desde el inicio, centrados en una visualización básica pero atractiva, sin procesamiento intensivo o análisis, y utilizando otra fuente de datos de base.

Esta aplicación utiliza la API (Application Programming Interface) de Wikia, una herramienta que proporciona datos del estado actual de la wiki como páginas, actividad reciente, artículos, usuarios, etc. La gran diferencia respecto a los datos del proyecto que recoge esta memoria es el hecho de que estos datos son únicamente los del estado actual, mientras que en los data-dumps se obtiene también información histórica de la wiki desde su nacimiento.

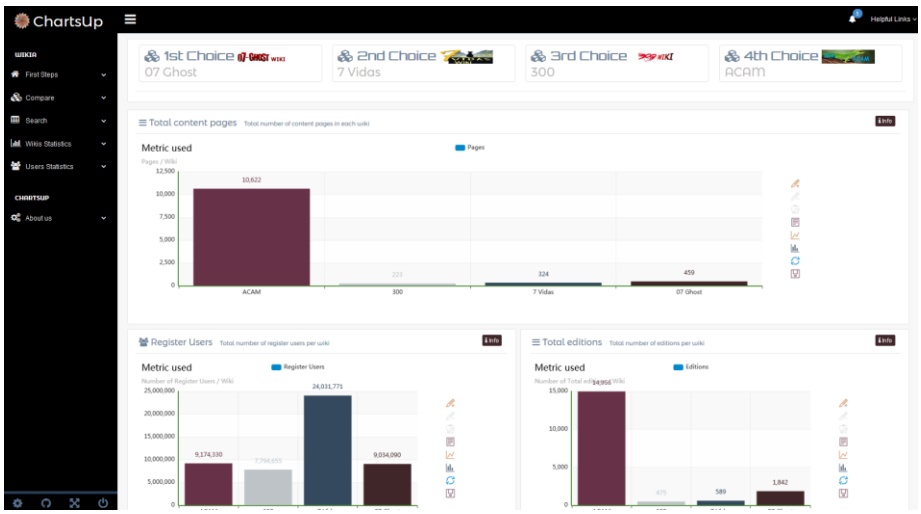


Figura 6. ChartsUp

Esta herramienta resulta muy útil a la hora de comparar wikis en la actualidad, mostrando algunas métricas sencillas como las páginas totales, las ediciones o los usuarios registrados. Sin embargo, a la hora de estudiar la evolución de una wiki, o tratar de analizar su comportamiento o tendencias, no resulta adecuada.

En contraste, el proyecto del TFG de esta memoria tiene un enfoque orientado hacia el análisis y procesamiento de la evolución de una wiki. Esto incluye la búsqueda de correlaciones entre las magnitudes que ChartsUp muestra en la actualidad (pero históricamente), o la monitorización del estado (en cualquier momento del tiempo).

4. METODOLOGÍAS DEL PROYECTO

Para desarrollar este proyecto han sido utilizadas distintas metodologías que han hecho posible obtener un resultado satisfactorio. A continuación, se explicarán en detalle junto con su aportación al proyecto.

4.1. Organización del trabajo

Este proyecto ha sido realizado por una única persona. Esto ha tenido implicaciones tanto positivas como negativas. La principal ventaja era la facilidad en la organización del tiempo, no tener que hacer reuniones de equipo y la autonomía. En cambio, se perdía el paralelismo de tareas, la diversidad de opiniones y el apoyo del equipo.

La comunicación con los directores del proyecto ha tenido dos medios. Por un lado, hubo una comunicación constante por medio de correo electrónico para estar en todo momento al tanto del estado del desarrollo. Además de esto, a lo largo de todo el curso ha habido reuniones periódicas con los tutores para ir definiendo el rumbo del proyecto y corregir los prototipos ya hechos. En un principio estas reuniones tenían lugar cada tres semanas, pero a medida que se fue acercando el final del curso se optó por reducir el periodo a dos semanas. Este feedback constante resultó fundamental para mantener un paso firme hacia un buen resultado.

Realizar una planificación desde el inicio no era tarea fácil, principalmente porque no se sabía realmente el camino que iba a seguir el trabajo. Así, las tareas se fueron definiendo en pequeños periodos según se encaminaba el proyecto.

4.2. Desarrollo iterativo incremental

El desarrollo de este proyecto se ha hecho de forma iterativa a lo largo de todo el periodo de trabajo. Aunque no se ha hecho una distinción claramente marcada de las distintas iteraciones, se pueden distinguir cuatro etapas claras.

1. **Brainstorming**. En un primer momento lo más importante era definir el camino que debía tomar el proyecto. Al ser un campo tan amplio las alternativas eran numerosas. Existía la posibilidad de continuar el trabajo realizado por compañeros del curso pasado [26] [27] o explorar una vertiente distinta. Una vez concluida esta etapa la decisión fue comenzar algo de cero y aventurarse a ver lo que se podía conseguir. Esta decisión fue basada en que el resultado del trabajo anterior no había resultado lo suficientemente satisfactorio para el análisis profundo de las wikis, además de que se quería intentar hacer un trabajo original en la medida de lo posible.
2. **Obtención de los datos.** Durante esta etapa se exploraron distintas posibilidades de fuentes de datos sobre las comunidades colaborativas estudiadas. La decisión fue optar por los logs de actividad de la plataforma, conocido como *data-dump.* Esta opción fue elegida en detrimento como otras como la API proporcionada por Wikia (sobre todo por la alta latencia de la obtención de datos) o el uso de un

crawler (con el que solo podemos obtener los datos que hay en la web, mientras que el dump nos ofrece información no visible).

El siguiente paso era analizar la estructura de estos data-dumps para ver qué información reveladora se podía extraer de ellos y transformarlos a un formato más manejable.

3. **Estudio de los datos.** Una vez extraídos los datos había que analizar qué información realmente se podía extraer. Para ello se inició el proceso de visualización de los datos usando Python. Durante esta iteración se generaron multitud de gráficas para ir aceptando o descartando cuales de ellas eran relevantes a la hora del estudio de la evolución de la wiki y la búsqueda de causalidad entre las distintas magnitudes (número de usuarios, número de páginas, número de ediciones, etc.).
4. **Interfaz de usuario.** Con las gráficas ya seleccionadas y la información útil para el análisis del desarrollo de las wikis ya seleccionada, lo siguiente sería crear un interfaz con el que los usuarios de la aplicación pudieran interactuar para realizar sus estudios. Resultaba de gran importancia para el resultado final la posibilidad de que el usuario pudiera interactuar con las distintas gráficas.

4.3. Visualización de datos

La visualización de datos es la presentación de los datos en un formato gráfico. Sin embargo, una visualización eficaz requiere mucho más que plasmar los datos. Actualmente, con el auge del Big Data y el análisis de datos, la representación gráfica de la información está cada vez más extendida, siguiendo la idea de que el cerebro humano procesa más fácilmente imágenes dinámicas que una lista de números. Grandes medios de comunicación como The Guardian o The New York Times han invertido mucho en este sector para elevar la capacidad de transmitir de sus noticias [28].

Sin embargo, una visualización eficaz no consiste únicamente en plasmar datasets (conjuntos de datos) sin ningún tipo de elaboración, existen multitud de factores que pueden incluso hacer que consigamos el efecto contrario al deseado. En un estudio del grupo Seeing Data [29] han identificado distintos factores que afectan esta capacidad de comunicación y que van más allá de los elementos visuales, como el tema que representan, el medio mediante el cual se presentan las gráficas o incluso las creencias y opiniones del propio receptor.

A pesar de esto, los elementos visuales representan un papel fundamental. A la hora de representar datos hay que tomar muchas decisiones sobre estos elementos. Algunos de ellos son los siguientes [30]:

- **El formato**. La interacción suele resultar siempre una ventaja para el usuario, pero dependiendo del medio puede resultar interesante la representación estática, como por ejemplo en un periódico.

- **El tipo de gráfica**. Esta quizá es la decisión más difícil, porque la representación unos mismos datos puede cambiar radicalmente. Muchos de ellos son conocidos, como el diagrama de tarta, de línea o de barra, pero existen multitud de tipos que pueden resultarnos útiles. Andy Kirk [31], especialista de Visualizing Data [32] estima que existen aproximadamente 75 tipos comunes, pero existen muchos más [33].
- Una vez decidido el tipo de gráfica existen otros aspectos muy importantes como **las variables a representar** o **la escala** en la que representarla. Usualmente se utiliza una escala linear, pero la escala logarítmica resulta bastante interesante para datasets con máximos y mínimos muy distanciados.

En lo que respecta a este proyecto, la visualización es quizá la parte más fundamental. La idea es transmitir información valiosa y reveladora sobre las wikis de Wikia, de manera que se puedan extraer conclusiones interesantes para el estudio de las mismas. Gracias a este mecanismo, investigadores con un perfil menos técnico pueden notar patrones y tendencias sin tener que indagar en el contenido en sí de los datos.

4.4. Software libre

Esta herramienta ha sido desarrollada como software libre. Por la propia naturaleza del estudio, resulta especialmente importante permitir que cualquiera profundizar y refinar más el código de la aplicación. Puede resultar de especial relevancia en el mundo académico, y gracias a que es software libre, tanto investigadores como desarrolladores podrán usarla como base para realizar sus estudios. Como ya se ha mencionado con anterioridad, el proyecto está públicamente en GitHub al alcance de cualquiera que lo desee. El enlace al repositorio es el siguiente:

https://github.com/Grasia/wikia_dashboard

El proyecto tiene una licencia MIT [34]. Este tipo de licencia garantiza las libertades del software libre (de uso, de modificación, de distribución y de distribución modificado) además de algunas otras libertades. La licencia permite que el software sea utilizado también como parte de software propietario o licenciar cambios con licencias más restrictivas.

5. TECNOLOGÍAS

El desarrollo de este proyecto trae consigo el uso de distintas tecnologías, algunas de ellas de uso muy extendidas en el desarrollo de aplicaciones. La principal de ellas es el lenguaje Python, utilizado en toda la aplicación, del que además se utilizan algunos módulos fundamentales para este proyecto. A su vez, estos módulos utilizan otras tecnologías comunes como las bases de datos relacionales o los lenguajes de desarrollo web HTML y CSS.

5.1. Python

Python [35] es un lenguaje de programación de alto nivel desarrollado por Guido van Rossum [36] en 1991. Entre otras, algunas de sus características principales son la legibilidad del código, gracias a que usa la sangría en para la separación de bloques en lugar de caracteres, los tipos dinámicos, la orientación a objetos y la reducción de líneas de código. Dispone de una gran cantidad de módulos activamente mantenidos en gran parte gracias a que es software libre [37].

Aunque quizá otros lenguajes como R [38] o Matlab [39] parezcan más adecuados para el tratamiento de datos, para construir una aplicación web interactiva y con un interfaz gráfico amigable Python ofrece más posibilidades. Además, gracias a librerías como NumPy [40], esta brecha entre lenguajes se reduce considerablemente.

De cara a el parseo de los datos, Python ofrece liberías que permiten de forma muy sencilla procesar documentos XML, que es el formato en el que vienen los data-dumps utilizados. Además, existen múltiples módulos de visualización de datos sencillas de usar, entre los que se encuentra el elegido para este proyecto: Bokeh.

Otro factor que se tuvo en cuenta fue la similitud con los lenguajes de programación de propósito general, lo que agilizó el aprendizaje de este nuevo lenguaje. Por todo esto Python ha sido elegido como lenguaje para el desarrollo de la herramienta aquí presentada.

Para el desarrollo en sí, no se utilizó un IDE complejo porque se estimó que no era necesario. En cambio, la herramienta utilizada fue el editor de texto Sublime Text [41], que permite ejecutar código Python en su consola, por lo que cubría las necesidades de este proyecto.

5.1.1. NumPy

Esta librería [42] ha sido utilizada sobre todo por las facilidades que ofrece en el manejo de estructuras vectoriales y matriciales numéricas. En particular, la conversión de estructuras de datos a vectores para poder realizar operaciones ha sido de gran ayuda a la hora de tratar los datos.

5.1.2. SQLite3

SQLite3 [43] [44] es un módulo que proporciona una implementación sin servidores de una base de datos SQL (structured query language) [45]. Aunque en un inicio no fue utilizado, a medida que el proyecto fue avanzando se hizo patente que la estructura de los datos tratados se adaptaba perfectamente al modelo relacional. Esta decisión fue tomada por dos motivos principales:

- En primer lugar, la estructura de los datos una vez procesados se ajustaba perfectamente a la de una tabla relacional.
- Al preparar los datos para las gráficas, se hizo patente que las operaciones que hacía falta realizar a los datos eran aquellas típicas de las bases de datos relacionales. Sobre todo, hablamos de las agrupaciones (la cláusula GROUP BY). Es una operación muy utilizada en la aplicación y resultaba tedioso, y seguramente ineficiente, hacer manualmente los bucles que hacían falta para realizarlas.

El almacenamiento de los datos utilizando este módulo se hace en un fichero (uno por base de datos), lo que resulta bastante conveniente para aplicaciones que se ejecutan en servidores locales.

5.1.3. Bokeh

Bokeh [46] es una librería para la visualización de datos de forma interactiva. La visualización se realiza a través de un interfaz web, utilizando su librería de JavaScript BokehJS [47]. En este proyecto se utiliza la arquitectura de servidor que proporciona Bokeh [48] para poder tener un grado de interacción mayor con el usuario.

La elección de esta librería ha sido debido a la amplia gama de gráficas y otros elementos relacionados que incluye, la facilidad de uso en Python y su ya mencionada interfaz web. Además, dado que la herramienta está orientada al análisis de comunidades colaborativas, resulta especialmente útil el hecho de que las gráficas se puedan guardar como imágenes.

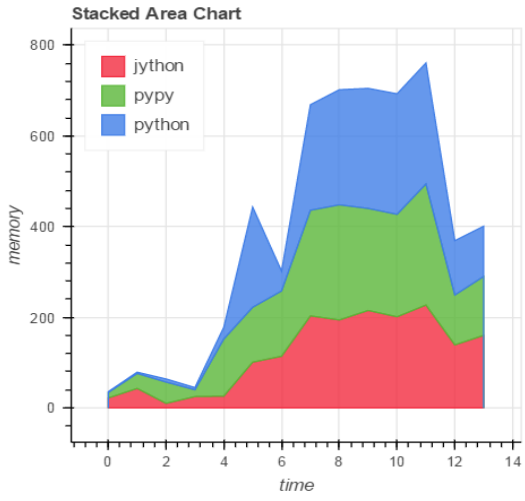


Figura 7. Ejemplo de visualización con Bokeh

5.2. HTML

Este lenguaje de marcado es la base estructural de la gran mayoría de las páginas que hay en Internet. En particular, Bokeh hace uso este lenguaje para generar la plantilla donde se introduce las gráficas y los scripts generados. Además, dependiendo del tipo de elemento y su disposición, se le asigna una clase u otra que más adelante será utilizada para las modificaciones de diseño.

En la arquitectura de servidor de Bokeh, se puede hacer uso de una plantilla HTML en la que encajar las gráficas generadas, lo que permite dar un aspecto más personalizado al resultado. A continuación, podemos observar un ejemplo de plantilla básica.

```
<!DOCTYPE html>
<html>
                              <head>
        <meta>
        <title>{{ title }}</title>
        {{ bokeh_head }}
    </head>
    <body>

        <h1>Título</h1>
        <p>Párrafo </p>

        {{ document }}

                              </body>
</html>
```

Figura 8. Estructura de la plantilla HTML para Bokeh

5.3. CSS

CSS (Cascading Style Sheets) [49] permite personalizar el aspecto de los elementos HTML para hacer el interfaz mucho más amigable. Bokeh hace uso de esta tecnología para la disposición de los elementos en los documentos, el ajuste del tamaño y todo aquello que suponga una modificación del aspecto final. Para ello los elementos disponen de clases propias de Bokeh.

Además, si se desea aumentar el grado de personalización, se pueden incluir anotaciones CSS dentro de la plantilla antes mencionada, siempre dentro de etiquetas *style.*

5.4. Sublime Text 3

Este editor de texto ha sido utilizado para el desarrollo del proyecto por distintos motivos. En primer lugar, puesto que su uso está bastante extendido ya estaba familiarizado con Sublime [41]. Su facilidad para navegar entre ficheros y carpetas, la facilidad de uso de todas sus herramientas, junto con la capacidad de poder compilar y ejecutar código en Python de manera sencilla hace que sea una buena elección para este proyecto.

5.5. Github

Github [50] es plataforma de desarrollo colaborativo de software que proporciona a sus usuarios repositorios con control de versiones Git [51]. Aunque en un proyecto en el que sólo trabaja una persona quizá esto puede resultar de menos utilidad, la principal baza de este servicio es el desarrollo colaborativo. Dada la naturaleza del proyecto, el interés que puede generar en las comunidades colaborativas y el potencial del tema resulta especialmente interesante exponer la herramienta al público para que su desarrollo sea más amplio.

En particular, el proyecto se encuentra alojado en el repositorio de GRASIA [52] [53]

6. LA FUENTE DE DATOS Y SU PREPROCESAMIENTO

6.1. La fuente de los datos

En primer lugar, era necesario decidir la fuente para obtener los datos a visualizar. Se vieron básicamente tres opciones para la extracción de los datos:

- La **API** de Wikia, que tenía una gran latencia a la hora de obtener los datos, por lo que fue descartada.
- Un **crawler** que obtuviera información directamente de las wikis. Esto no nos permitiría saber información del desarrollo de la wiki, puesto que sólo sabríamos su estado actual.
- La **exportación de la base de datos**, conocida como data-dump, que contiene un registro de toda la actividad que ha habido a lo largo de la wiki en todas las páginas, lo que permite ver detalladamente la evolución.

La decisión final fue la de analizar los data-dumps que proporciona la propia herramienta de Wikia y está a la mano de cualquiera. Un **data-dump** es un archivo que contiene **logs de toda la actividad que ha tenido lugar en una determinada wiki**.

Para conseguir un data-dump es necesario acceder a la página de estadísticas de la wiki que nos interese, lo que se hace añadiendo "/Special:Statistics" al final de la URL de la misma, por ejemplo: http://es.zelda.wikia.com/special:statistics. Antes de proceder a la descripción, es importante aclarar ciertos puntos sobre los data-dumps:

- Al acceder la página antes mencionada se nos ofrecen dos opciones: descargar la base de datos incluyendo las páginas en su estado actual únicamente o un historial completo que incluye incluso páginas que ya no forman parte de la wiki. Para tener una visión global de la evolución de la wiki se ha estado trabajando con los data-dumps incluyendo el historial.

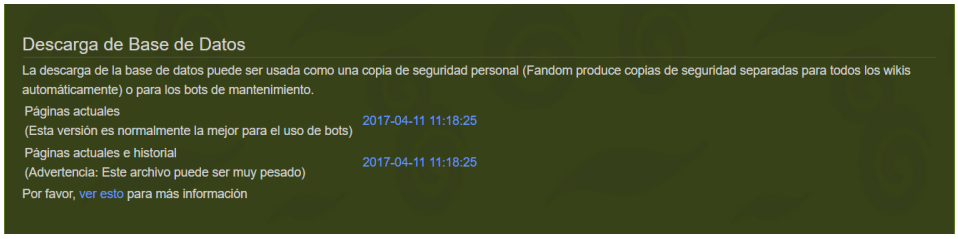


Figura 9. Ejemplo de sección de descarga de data-dump

- En algunos casos es posible que el data-dump no haya sido generado. Si es así, lo mejor será contactar con algún administrador de la wiki y solicitarle que genere el dump.
- En el desarrollo de este proyecto se ha detectado un bug de la herramienta generadora de data-dumps de Wikia. A partir de cierto tamaño de wiki (sin determinar), los data-dumps son generados corruptos o incompletos, lo que ha imposibilitado el estudio de wikis maduras.

Más información sobre la descarga de la base de datos de las wikis de Wikia se puede encontrar en la página correspondiente de ayuda [54].

6.2. Estructura del data-dump

Los data-dumps en sí son archivos XML de gran tamaño (mínimo cientos de MB), que contienen la información correspondiente a la actividad dentro de la wiki. La estructura del documento se compone de la información del sitio, seguido de un listado de todas las páginas de la wiki.

```
<mediawiki>
        <siteinfo> … </siteinfo>
        <page> … </page>
        .
        .
        .
        <page> … </page>
</mediawiki>
```

Figura 10. Estructura del data-dump

La información del sitio se compone del título, la dirección web (con IP), la versión de MediaWiki utilizada y una lista con los distintos namespaces (tipos de página, como página de contenido, página de foro, perfil de usuario…).

```
<siteinfo>
  <sitename>The Legend of Zelda Wiki</sitename>
  <base>http:///10.8.64.40/wiki/The_Legend_of_Zelda_Wiki</base>
  <generator>MediaWiki 1.19.9</generator>
  <case>first-letter</case>
  <namespaces>
    <namespace key="-2" case="first-letter">Medio</namespace>
    <namespace key="-1" case="first-letter">Especial</namespace>
    <namespace key="0" case="first-letter" />
    <namespace key="1" case="first-letter">Discusión</namespace>
    <namespace key="2" case="first-letter">Usuario</namespace>
    <namespace key="3" case="first-letter">Usuario discusión</namespace>
  </namespaces>
</siteinfo>
```

Figura 11. Ejemplo de etiqueta <siteinfo> (vista en Fig. 10) rellena

Por otro lado, cada una de las páginas tiene un título, un namespace, un identificador único, un hash sha1 y una lista de revisiones. Cada revisión tiene un identificador único, una fecha y hora, un usuario identificado bien por su id y nombre si es registrado o bien por su IP si es anónimo. Al final de cada revisión, el data-dump contiene el texto de la página tras la modificación realizada, así como el tamaño en bytes de la misma. Existen otros elementos como comentarios o los atributos "deleted" que no se han tratado en este proyecto.

Figura 12. Ejemplo de página con una única revisión

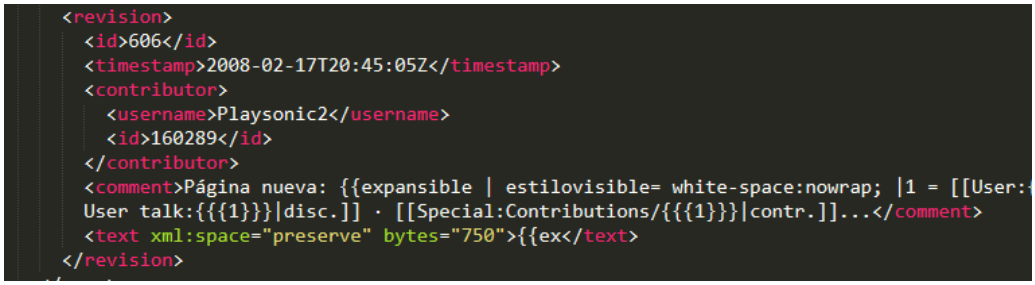


Figura 13.Ejemplo de revisión.

6.3. Preprocesamiento de los datos

Una vez comprendida la estructura de los datos era necesario parsearlos (traducir de su formato original) a un formato más manejable por dos motivos principalmente:

- En primer lugar, los data-dumps por su gran tamaño tomaban bastante tiempo de procesado, por lo que era inviable tratar directamente con ellos.
- Además, mucha de la información incluida en los archivos no aportaba demasiado valor de cara a nuestras necesidades, por lo que nos quedamos únicamente con lo que nos interesaba

Así, después de un procesado en el que se entrará más en detalle en el capítulo sobre la arquitectura, los datos reales con los que trabaja e interactúa la aplicación se redujeron. El resultado final es un fichero de texto en formato csv (datos en filas separando campos por un carácter, que en nuestro caso es el punto y coma) cuya primera fila es el título de la wiki, seguido de una fila para cada una de las revisiones que contiene la siguiente información:

- **ID de la página**
- **Título de la página**
- **Namespace de la página.**
- **ID de la revisión.**
- **Timestamp de la revisión.**
- **ID del autor.** En caso de autores anónimos este campo sería la dirección IP.
- **Nombre del autor.** En caso de autores anónimos este campo tendría el valor "Anonymous".
- **Tamaño de la página.**

Ésta sería la estructura de los datos obtenidos del script de parseo de los archivos XML. Sin embargo, a la hora de insertar los datos en la base de datos que utiliza la aplicación, se han realizado el cálculo de una serie de valores derivados para optimizar el tiempo, ya de por sí extenso, que toman las distintas consultas. Como resultado, tendríamos las siguientes magnitudes nuevas:

- **Month_group:** para facilitar la agrupación por meses se almacena la fecha truncada al primer día del mes correspondiente al timestamp de la revisión.
- **Year_group:** similar a la anterior pero truncada al primer día del año.
- **Creation:** este "flag" se utiliza para distinguir rápidamente aquellas revisiones que supongan la creación de una página nueva, es decir, la primera revisión correspondiente a una página. Así se pueden separar las revisiones en creaciones y ediciones.
- **Revision_bytes:** resulta interesante hacer un análisis no solo del tamaño final de la página después de la revisión, que nos lo proporciona directamente el data-dump, sino el número de bytes que ha afectado la revisión, bien porque hayan sido quitados o porque hayan sido añadidos.

  El valor de este atributo se calcula como la diferencia, en valor absoluto, del número de bytes de la página tras la revisión actual menos el tamaño tras última revisión. En caso de que sea la primera revisión, es decir la creación, se considera el tamaño de la revisión igual al tamaño de la página.

Una vez pre-procesados los datos y transformados en un formato más manejable, se puede continuar con su transformación y análisis, explicado en las siguientes secciones.

7. ARQUITECTURA DE LA APLICACIÓN

A continuación, se explicarán en detalle los aspectos técnicos relacionados con el proyecto. En particular, se explicará la función de cada uno de los ficheros, la estructura de la base de datos y la arquitectura de servidor utilizada.

La aplicación consta de tres módulos: el parser de los data-dumps, el gestor de la base de datos y el main, que se encarga de la generación de las gráficas. Cada uno de ellos realiza una parte fundamental del proceso que será necesaria en el flujo de la aplicación. El siguiente esquema refleja a grandes rasgos la relación entre los distintos módulos:

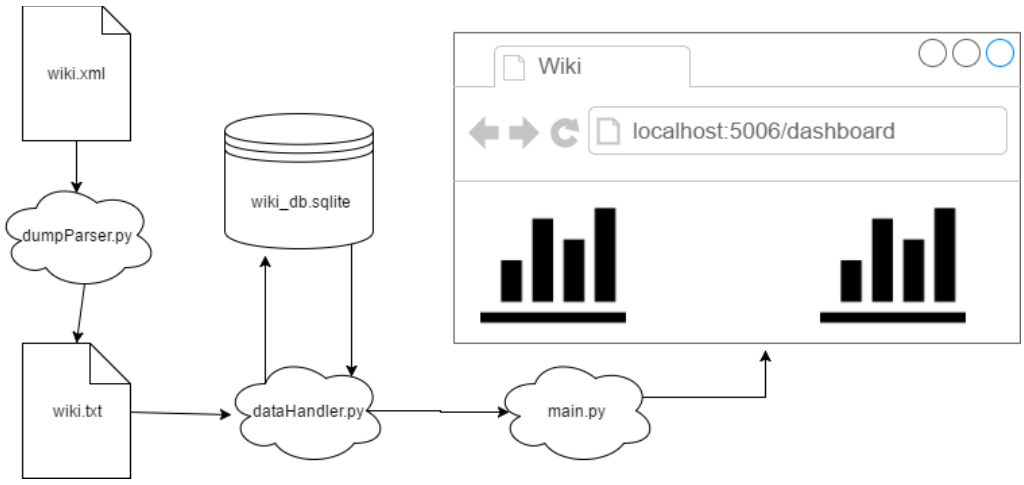


Figura 14. Esquema de las relaciones entre los distintos elementos de la aplicación.

Los datos de los que parte la aplicación son los proporcionados por el data-dump, un fichero en el formato XML descargado de la página de estadísticas de la wiki. Este fichero es procesado por el script ***dump_parser.py***, que genera un archivo de texto como salida. Este archivo de texto es el que utiliza el script ***dataHandler.py,*** que se encarga tanto de insertar los datos en la base de datos, como de realizar las consultas a la misma. El resultado de estas consultas es utilizado por el script ***main.py***, que genera las gráficas que posteriormente Bokeh se encarga de volcar en el navegador. Los tres archivos Python han sido desarrollados en este TFG y forman el núcleo de la aplicación.

7.1. Parseo de los datos

Como ya se explicó en el capítulo sobre el preprocesado de datos, la información en bruto con la que se trata proviene de un archivo en formato XML, por lo que era necesario procesarla y transformarla para que fuera más manejable. Para ello se creó un script que se encargaba de esta operación: el fichero ***dump_parser.py***.

En un primer momento se eligió una librería para Python llamada Untangle [55], que ofrece la funcionalidad de procesamiento de archivos XML en este lenguaje. La decisión de usar esta librería se tomó gracias a su facilidad de uso y a que resultaba similar a librerías de similar funcionalidad para otros lenguajes.

Sin embargo, llegado el punto de poner a prueba la herramienta con wikis de tamaño considerable se descubrió que el parser no funcionaba debido a un error de memoria. Después de investigar se descubrió que el motivo de este error era que, a la hora de procesar el archivo, la librería cargaba el documento entero en memoria. Dado el gran volumen de datos con el que trata este proyecto, esa carga en memoria sobrepasaba la capacidad de una máquina estándar.

La solución fue realizar un recorrido iterativo a lo largo del documento. Para ello fue necesario cambiar a la librería ElementTree XML de Python [56] incluida en el paquete estándar del lenguaje. Así, gracias a su función de parseo iterativo fue posible procesar archivos de cualquier tamaño.

El proceso de parseo en sí es relativamente simple: recorrer el archivo en busca de las etiquetas que tenían la información deseada (siguiendo la estructura XML ya explicada). Las únicas medidas especiales que hacía falta tomar era distinguir entre usuarios anónimos y registrados (era necesario asignar un nombre a los anónimos para tener unos datos homogéneos), evitar caracteres especiales que se utilizan como separador (en particular el punto y coma) y algunas etiquetas con el atributo “deleted”. Como resultado se obtiene un archivo de texto (“.txt”) con la siguiente estructura:

- El nombre del fichero de salida es el mismo que el del fichero de entrada, salvando la extensión.
- La primera línea es el título de la wiki.
- Una segunda línea de cabecera que sirve para explicar qué significa cada campo en las líneas.
- Una línea o registro por cada una de las revisiones con el siguiente formato:

[page_id;page_title;page_ns;revision_id;timestamp;contributor_id;contributor_name;bytes]

Este script es independiente y se puede ejecutar independientemente de la aplicación principal. Su uso es explicado en el anexo “manual de usuario”.

7.2. Almacenamiento y acceso a los datos

Una vez transformados los datos, el siguiente paso era almacenarlos, procesarlos y proporcionar una capa de acceso a los mismos. Ésta es la función del fichero ***dataHandler.py.*** Al inicio, los datos se extraían directamente del fichero de texto y se iban manejando con distintas estructuras de datos (diccionarios, listas, etc.). Sin embargo, a medida que avanzaba el proyecto quedaba cada vez más claro que tanto los datos como las operaciones que se realizaban con ellos se adaptaban perfectamente a una base de datos tradicional.

En particular se utiliza SQLite, una implementación de base de dato relacional que se almacena en un fichero local. Puesto que la herramienta se ejecuta en modo local esta opción resultaba la más adecuada. Al fichero utilizado como base de datos por la herramienta se le ha dado el nombre de ***wikis_db.sqlite.*** La base de datos consta de dos tablas:

- **Wikis:** esta tabla se ha creado para ofrecer la posibilidad de almacenar la información de varias wikis en la misma base de datos.

Columna	Tipo	Descripción
Id	Integer	Identificador único autogenerado
Name	Text	Nombre de la wiki

- **Revisions:** almacena los registros de las revisiones de las distintas wikis.

Columna	Tipo	Descripción
Id	Integer	Identificador único autogenerado
Wiki_id	Integer	Identificador de la wiki, que es clave ajena de la tabla wikis.
Page_id	Integer	Identificador de la página
Page_title	Text	Título de la página
Page_ns	Integer	Namespace (o tipo) de la página
Revision_id	Integer	Identificador de la revisión único en toda la wiki.
Revision_date	Timestamp	Fecha y hora de la revisión
Contributor_id	Text	ID del autor o IP en caso de autores anónimos
Contributor_name	Text	Nombre del autor o “Anonymous” en caso anónimo.
Page_size	Integer	Tamaño de la página después de la revisión
Revision_bytes	Integer	Bytes que ha cambiado la revisión.
Month_group	Timestamp	Fecha de la revisión truncada al mes.
Year_group	Timestamp	Fecha de la revisión truncada al año.
Creation_group	Integer	Flag que vale 1 si es la primera revisión de la página.

La carga de datos está hecha en varias etapas puesto que no basta con lo que se saca de los ficheros de texto. A continuación, se describirá el proceso de carga:

- Se lee el archivo de texto.
- Se borra toda la información que haya sobre esa wiki y se inserta en la tabla wikis, devolviendo su ID.
- Se insertan los datos obtenidos del archivo de texto.
- Se actualizan los campos month_group y year_group con el campo revision_date insertado en el paso anterior.
- Se calculan los bytes de la revisión (restando, el valor absoluto, el tamaño de la página tras la revisión actual menos el tamaño tras la revisión anterior) y se ponen los flags de creación (la revisión con la fecha más baja para esa página).

Una vez finalizado el proceso, los datos están listos para que el generador de gráficas pueda acceder a ellos. El acceso a los datos se hace a través de los métodos del *dataHandler* que se encargan de hacer distintas consultas. En general, puesto que la aplicación trata con datos a lo largo del tiempo, casi todas las consultas devuelven diccionarios que tienen para cada una de las fechas la información correspondiente. Esto hace posible que el usuario pueda interactuar y elegir las fechas.

Existen múltiples métodos con sus correspondientes consultas, pero una de ellas es la principal. Se trata de la que obtiene los datos del estado general de la wiki a lo largo del tiempo. La mayoría de las gráficas, así como los paneles de estadísticas utilizan esta información. Es la consulta que más tiempo tarda en ejecutarse, pero aun así resulta más eficiente que realizar muchas consultas. A continuación, se muestra el código del método correspondiente.

```
def wiki_status(wiki):
    conn = sqlite3.connect(db_path)
    c = conn.cursor()
    c.execute("SELECT distinct(month_group) month from revisions where wiki_id=? and page_ns = 0 order by month asc ",(wiki,))
    dates = c.fetchall()
    result ={}
    for date in dates:
        c.execute('''SELECT sum(creation) as pages,
                            sum(case when creation = 0 then 1 else 0 end) editions,
                            count(distinct(case when contributor_name <> 'Anonymous' then contributor_id else null end)) as logged_users,
                            count(distinct(case when contributor_name = 'Anonymous' then contributor_id else null end)) as anonymous_users,
                            avg(case when creation = 1  then null else revision_bytes end) avg_revision_bytes
                    from revisions
                    where month_group <= ? and page_ns = 0 and wiki_id = ?''',[date[0],wiki])
        result[dt.strptime(date[0],"%Y-%m-%d")] = c.fetchall()
    conn.close()
    return result
```

Figura 15. Ejemplo de consulta SQL

7.3. Creación de las gráficas

El fichero principal es el main.py**.** Se encarga de generar los elementos de visualización de la aplicación, así como de todas las funcionalidades de interacción con el usuario.

7.3.1. Gráficas

Las gráficas suponen los principales elementos del interfaz gráfico puesto que son los que muestran los datos en sí. Se utilizan sobre todo diagramas de barras y de línea, dependiendo de si se muestra una evolución o actividad mensual no acumulada. A nivel de implementación cabe distinguir dos tipos: las figuras básicas y las gráficas de alto nivel.

Las figuras básicas aportan una libertad mucho mayor a la hora de visualizar los datos, pero con el inconveniente de que hay que configurar muchos parámetros. Esto permite salirse de los diagramas más convencionales o también trabajar con datos más complejos. La fuente de los datos en este tipo de gráficas es un *Column Data Source* (una clase de Bokeh utilizada para datos). Esto facilita, como explicaremos más adelante, la actualización de las gráficas al añadir interacciones.

Por otro lado, existen las gráficas de alto nivel. Se trata de diagramas ya configurados para tener formas estándar (barras, tarta, mapa de calor…). Es la forma más sencilla de representar con la librería utilizada, pero tiene sus desventajas. En primer lugar, las posibilidades de personalización son bastante bajas, pero el mayor problema es que la actualización dinámica de los datos resulta imposible. Esto implica que a la hora de hacer interacciones estas gráficas hay que volver a generarlas con nuevos datos.

```
pages_figure = figure(  title = "Total pages",
                        name ="Total pages",
                        y_axis_label = "Pages",
                        height = HEIGHT,
                        width = WIDTH,
                        x_axis_type = "datetime",
                        tools = tools)
pages_figure.line(  'dates',
                    'pages',
                    source=wiki_status_source,
                    line_width=2.5,
                    color =Paired9[3])
hover = pages_figure.select(dict(type=HoverTool))
hover.tooltips = [
('Date','@time'),
('Total Pages','@pages')
]
```

Figura 16. Ejemplo de creación de gráfica.

7.3.2. Widgets

En esta librería, los *widgets* son pequeños elementos auxiliares utilizados para complementar aquellos puramente de visualización. Los distintos elementos utilizados han sido los siguientes:

- **Slider:** un *slider* es una barra deslizante que permite seleccionar un valor dentro de un rango. En nuestra aplicación es utilizado en la pestaña de estadísticas para elegir el periodo del que se quieren ver los datos. Los principales parámetros de configuración son: el rango de valores, el valor inicial, así como el salto que hay entre valores.

```
time_slider = Slider(start=1,
                     end= len(dates),
                     value= len(dates),
                     step = 1,
                     title = "Months from creation",
                     width=1580)
```

Figura 17. Creación del Slider

- **CheckBoxGroup:** se trata de cajas para poder seleccionar distintas opciones. En este proyecto han sido utilizados en la pestaña de evolución para permitir al usuario seleccionar las gráficas que desea mostrar en pantalla. Para crearlo es necesario definir las etiquetas de las distintas opciones y aquellas que al inicio están seleccionadas.

```
cbg_labels = [  "Total pages",
                "Monthly new pages",
                "Total pages per user",
                "Monthly edited pages",

                "Total editions",
                "Monthly editions",
                "Total editions per user",
                "Total editions per page",
                "Monthly editions per edited page",
                "Editions by author type",

                "Total users",
                "Total logged users",
                "Total anonymous users",
                "Monthly new users",
                "Monthly new logged users",
                "Monthly new anonymous users",
                "Active users by type",

                "Average page size",
                "Average edition bytes",
                "Monthly average edition bytes"]
cbg_pages = CheckboxGroup(labels=cbg_labels[0:4],active=[0], css_classes =['pages_checks'])
cbg_editions =CheckboxGroup(labels=cbg_labels[4:10],active=[0], css_classes =['edition_checks'])
cbg_users = CheckboxGroup(labels=cbg_labels[10:17],active=[], css_classes =['users_checks'])
cbg_ratios=CheckboxGroup(labels=cbg_labels[17:20],active=[], css_classes =['other_checks'])
```

Figura 18. Creación del CheckBoxGroup

- **Button:** el único botón utilizado en la aplicación sirve para quitar todas las gráficas mostradas en la página de evolución.
- **Div:** este elemento de Bokeh es utilizado para embeber elementos en HTML puro. La aplicación lo utiliza para mostrar información en texto plano, como los títulos y los rótulos.

La mayoría de estos elementos suponen una interacción con el usuario, y por ello es necesario definir las acciones a realizar en caso de interacción. Además, se puede modificar su aspecto utilizando clases css.

7.3.3. Interacciones

Como se ha mencionado, algunos widgets requieren que suceda algo cuando el usuario haga alguna interacción. Estas interacciones vienen definidas por funciones llamadas *callbacks*, que se llaman cuando se activa algún evento (cambio, *click*…). Estos *callbacks* pueden ser código Javascript, que se usa fundamentalmente cuando los programas devuelven un archivo HTML, o código Python, que se utiliza en modo servidor como es nuestro caso. Algunos ejemplos de los *callbacks* utilizados son:

- ***Callback* del *slider***: se encarga de actualizar toda la información mostrada en la pestaña de estadísticas. Es activado con evento de cambio. Como parámetros tiene el atributo que ha sido modificado y sus valores antes y después del cambio.

```
time_slider.on_change('value',slider_callback)
```

Figura 19. Asignación del gestor de eventos al Slider

```
def slider_callback(attr,old,new):
    banners_div.text = banners_html(new)
    date_div.text = '<h1 style="text-align:center">'+time[new-1]+'<h1>'
    top_users_source.data = top_users_table_ds[dates[new-1]].data
    top_pages_source.data = top_pages_ds[dates[new-1]].data
    update_hm(dates[new-1])
```

Figura 20. Gestor de eventos del Slider

- ***Callback* de los *checkbox*:** se encarga de mostrar aquellos que están seleccionados y ocultar todos los demás. Para mejorar la experiencia de usuario, los *checkbox* se han agrupado por temáticas. Esto implica que hay que hacer un paso previo de unificar la información de todos antes de proceder a mostrar y ocultar gráficas. Es activada cuando se pincha en alguna de las opciones.

```
cbg_pages.on_click(cb_callback)
cbg_editions.on_click(cb_callback)
cbg_users.on_click(cb_callback)
cbg_ratios.on_click(cb_callback)
```

Figura 21. Asignación de los gestores de evento a los CheckBox

```
def cb_callback(active):
    active1 = cbg_pages.active
    active2 = [element + 4 for element in cbg_editions.active]
    active3=[element + 9 for element in cbg_users.active]
    active4=[element + 17 for element in cbg_ratios.active]
    active = active1 + active2 + active3 + active4
    rootLayout = curdoc().get_model_by_name('graphs')
    children = rootLayout.children
    for i in range(0,len(all_figures)):
        plt = curdoc().get_model_by_name(all_figures[i].name)
        if i in active:
            if not plt:
                children.insert(0,all_figures[i])
        else:
            if  plt:
                children.remove(plt)
    for fig in children:
        fig.x_range = children[0].x_range
```

Figura 22. Gestor de evento de los CheckBox

7.4. Arquitectura de servidor de Bokeh

La arquitectura de servidor de Bokeh ha sido utilizada para ofrecer una mayor interacción de los usuarios con las gráficas. En nuestra aplicación, el servidor desplegado se hace de forma local. El despliegue se hace a partir de código Python, generando distintos documentos para cada una de las sesiones abiertas en distintas navegadores o pestañas (no se mantiene la sesión entre pestañas del navegador).

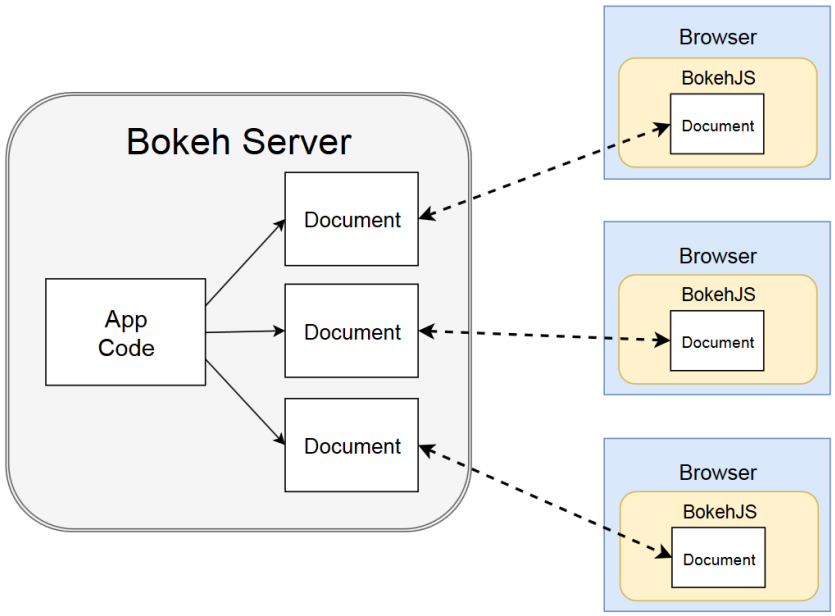


Figura 23. Arquitectura de servidor Bokeh

Las aplicaciones de servidor de bokeh consisten en un fichero main, que contiene el código principal del programa. Además, existen otros elementos opcionales que aportan distintos recursos. En este proyecto la estructura de ficheros es la siguiente:

```
dashboard
   |
   +---dataHandler.py
   +---main.py
   +---dump_parser.py
   +---wikis_db.sqlite
   +---static
   |    +---anonym_user.png
   |    +---logged_user.png
   |    +---editions.png
   |    +---pages.png
   |    +---fandom.png
   |    +---up.png
   |    +---down.png
   |    +---equal.png
   |
   |---templates
   |    +---index.html
   |
   +---theme.yaml
```

Además de los scripts de Python ya explicados, tenemos algunos ficheros extra:

- **wikis_db.sqlite:** es la base de datos mencionada en el apartado sobre el dataHandler.
- **theme.yaml**: es un fichero donde se define el valor de algunos atributos genéricos para los elementos de Bokeh.
- **Templates**: este directorio contiene la plantilla HTML donde se incrustarán los elementos generados por Bokeh.
- **Static:** en esta carpeta se guardan los elementos estáticos, como son las imágenes o los archivos de estilo.

8. INTERFAZ DE LA APLICACIÓN

La aplicación consta de dos pestañas, una para analizar la evolución de distintas magnitudes y otra para analizar el estado de la wiki en un determinado momento de su existencia. El título tanto dentro de la página como en la pestaña es el de la wiki que se está analizando. En un primer momento, la pestaña por defecto es la de evolución.

A continuación, se explicará en detalle el contenido de cada una de las pestañas.

8.1. Evolution

La pestaña de evolución está pensada para que el usuario pueda comparar distintas magnitudes de la wiki a lo largo del tiempo, buscando correlaciones o causalidades. Está dividida en dos partes. La primera de ellas es un panel donde se pueden seleccionar las distintas gráficas que se desea ver. La otra parte es donde realmente se visualizan las gráficas seleccionadas.

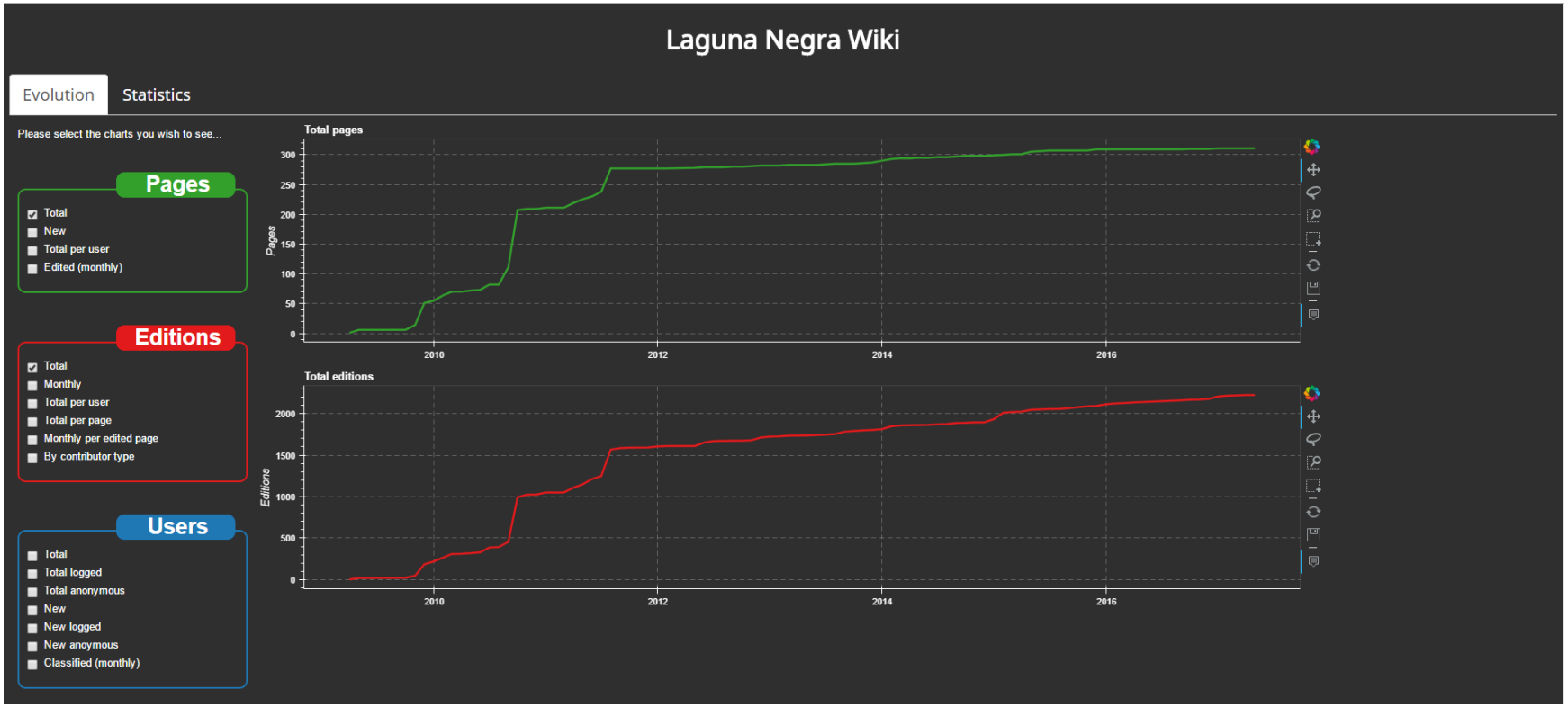


Figura 24. Ejemplo de pestaña de evolución

En el panel de la primera pestaña, las opciones están agrupadas temáticamente en 4 conjuntos para facilitar la búsqueda por parte del usuario. Además, se utiliza un código de colores tanto en el panel como en las gráficas para que el usuario pueda identificar sin esfuerzo la información que se está mostrando. También se utilizan distintas tonalidades en función de si son magnitudes acumuladas o mensuales. En definitiva, los códigos visuales utilizados son:

- Color azul para la información relacionada con los usuarios.
- Color verde para la información relacionada con las páginas.
- Color rojo para la información de las ediciones.
- Color naranja para la información respectiva al tamaño tanto de las ediciones como de las páginas.

Además, para distinguir las magnitudes acumuladas de las magnitudes mensuales se ha hecho lo siguiente:

- Para las magnitudes acumuladas se usa diagramas de líneas porque representa todo lo que ha sucedido hasta ese momento. Los colores utilizados son de todos oscuros.

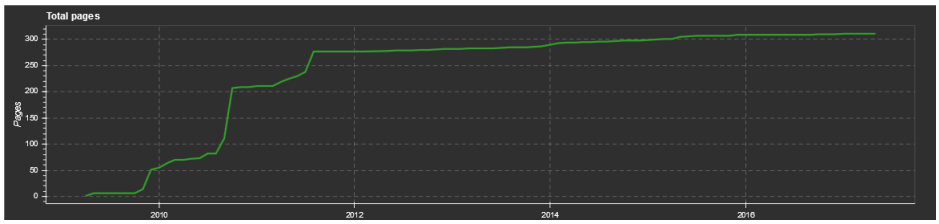


Figura 25. Ejemplo de gráfica acumulada

- En el caso de las magnitudes mensuales los diagramas son de barras y los colores de todos claros.

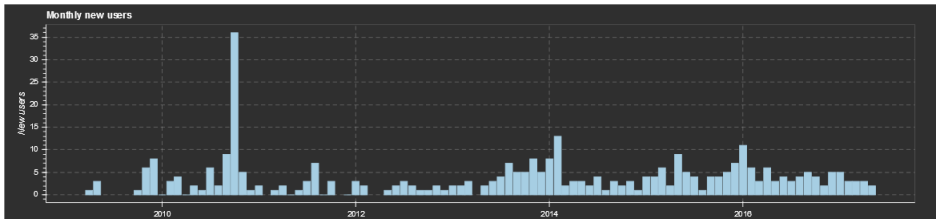


Figura 26. Ejemplo de gráfica acumulada

A continuación, se procederá a describir las distintas magnitudes que se pueden visualizar la aplicación, explicando cómo se calcula cada una de ellas.

- **Total pages:** muestra el número total de páginas de contenido hasta una determinada fecha.
- **Monthly new pages:** muestra las páginas que han sido creadas en un determinado mes.
- **Total pages per user:** se calcula dividiendo el número total de páginas entre el número total de usuarios (es decir, número de páginas hasta la fecha entre número de usuarios hasta la fecha) para así obtener un ratio de cuántas páginas hay por cada usuario.

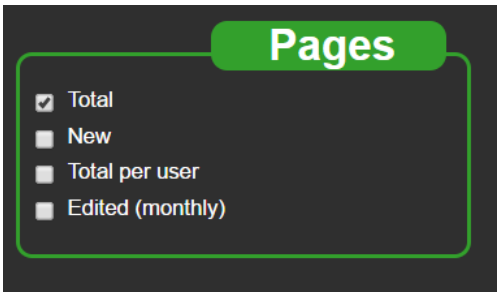


Figura 27. Panel de páginas.

- **Monthly edited pages:** es el número de páginas que han sido editadas en un determinado mes.
- **Total editions:** muestra el número total de ediciones en páginas de contenido que han sido hechas hasta una determinada fecha. Una edición se considera una revisión que no es creación, es decir, para cada página todas exceptuando la primera.
- **Monthly editions:** muestra el número de ediciones que han sido hechas en un determinado mes.
- **Total editions per user:** se calcula dividiendo el número total de usuarios entre el número total de usuarios. Así obtenemos una media de las ediciones por cada usuario.

- **Total editions per page:** se calcula dividiendo las ediciones totales entre el número total de páginas para obtener una media de las ediciones por cada página.
- **Monthly editions per edited page:** en cada mes se calcula dividiendo las ediciones que han sido hechas ese mes entre el número de páginas que han sido editadas ese mes.
- **Editions by author type:** son las ediciones ocurridas en ese mes, pero distinguiendo aquellas hechas por usuarios registrados y aquellas hechas por usuarios anónimos.

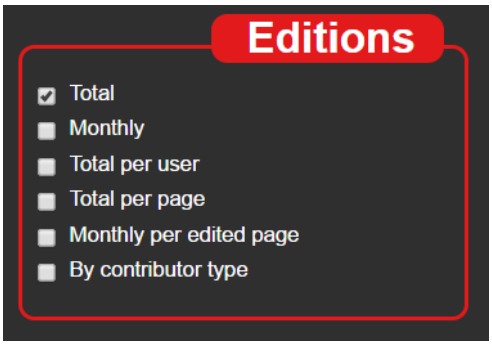


Figura 28. Panel de ediciones

- **Total users:** muestra el número total de usuarios hasta la fecha. Cabe destacar que, dado el origen de los datos, para este proyecto se considera usuarios a aquellos que han editado alguna vez.
- **Total logged users:** total de usuarios registrados que hay hasta la fecha.
- **Total anonymous users:** total de usuarios anónimos que hay hasta la fecha. Al no estar registrados, su único elemento distintivo es la dirección IP.
- **Monthly new users:** usuarios nuevos en ese mes, es decir, usuarios que nunca habían editado antes.
- **Monthly new logged users:** usuarios registrados nuevos.
- **Montly new anonymous users:** usuarios anónimos nuevos.
- **Active users by type:** usuarios que han editado ese mes divididos en tres tipos. Esta clasificación es propuesta por Felipe Ortega en su estudio sobre Wikipedia [14] y se ha considerado aplicable a las wikis de Wikia. Los tipos son los siguientes:
  - **Very active:** aquellos que han realizado más de 100 veces en ese mes.
  - **Active:** usuarios que han editado más de 5 pero menos de 100 veces ese mes.
  - **Other:** usuarios que han editado al menos una vez ese mes pero que no han llegado a las 5 ediciones.

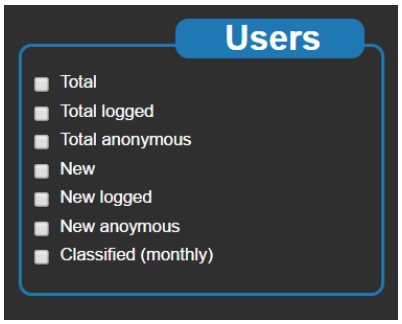


Figura 29.Panel de usuarios

- **Average page size:** se calcula haciendo la media de los tamaños de página de todas las páginas hasta la fecha.
- **Average edition bytes:** se calcula haciendo la media del tamaño de todas las ediciones hasta la fecha.
- **Monthly average edition bytes:** se calcula haciendo la media del tamaño de las ediciones de cada mes.

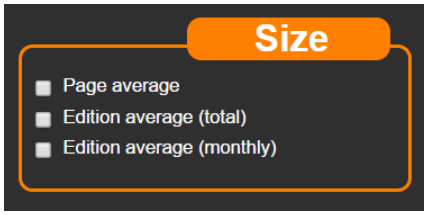


Figura 30. Panel de tamaño

Por último, se facilita un botón que permite quitar todas las gráficas para que el usuario no tenga que ir quitándolas una a una.

8.2. Statistics

La pestaña de estadísticas muestra el estado de la wiki en determinado punto del tiempo. Utilizando una barra deslizante para seleccionar el periodo en el que se quiere situar, el usuario puede observar distinta información de interés que le permita analizar la wiki. En información puede resultar de especial interés para que los administradores de las wikis puedan analizar la actividad de un determinado mes.

La pantalla tiene distintas secciones distintos métodos de visualización. Tenemos una barra de estado que nos da los valores de distintas magnitudes directamente, así como su variación respecto al mes anterior. Después tenemos un par de tablas con rankings tanto de usuarios como páginas y a su lado la distribución de las ediciones de ese mes representadas en diagramas de barras horizontales.

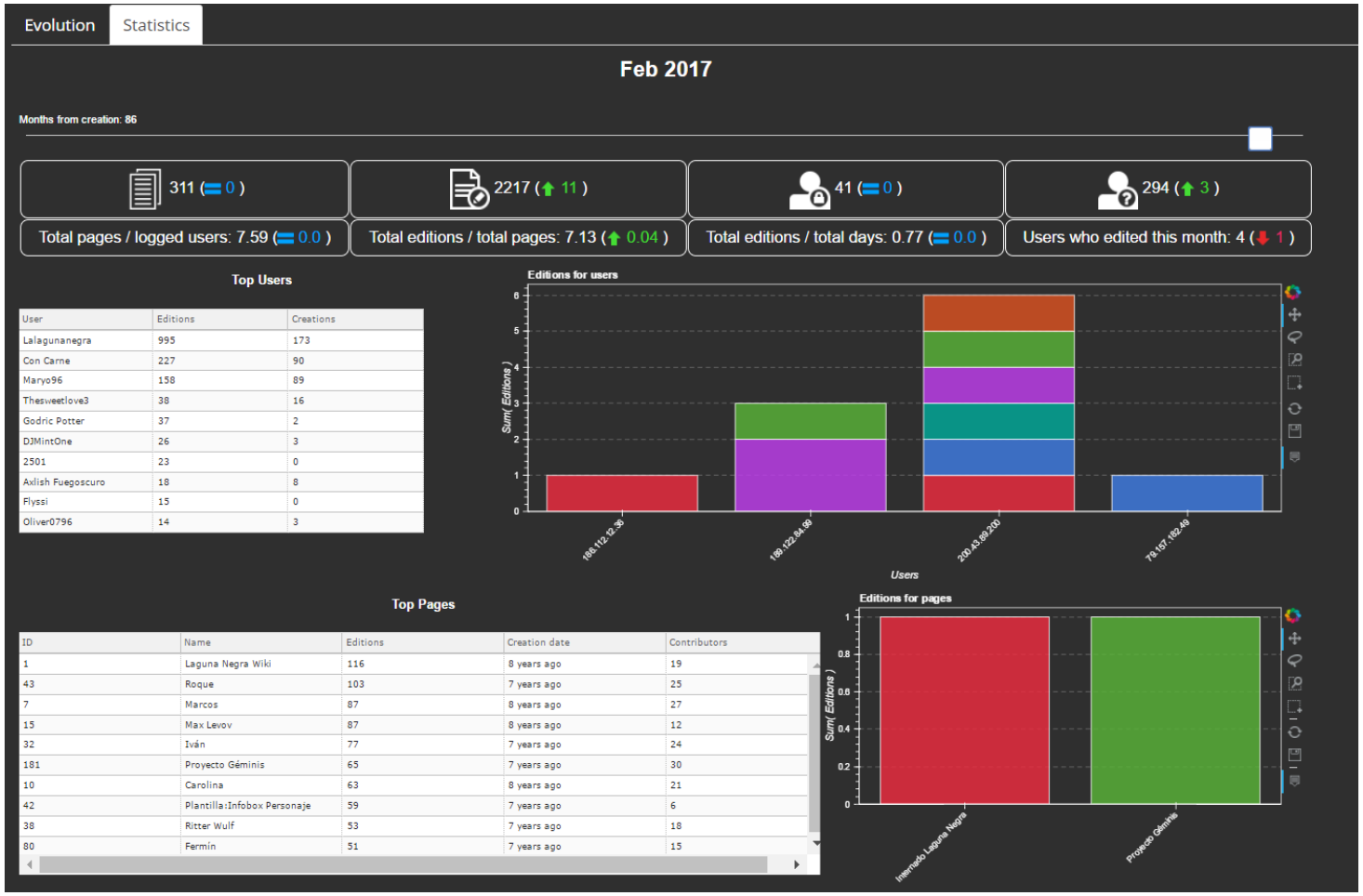
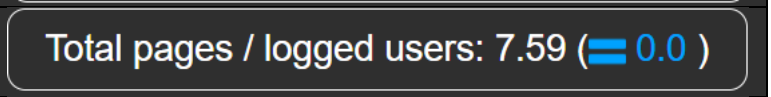
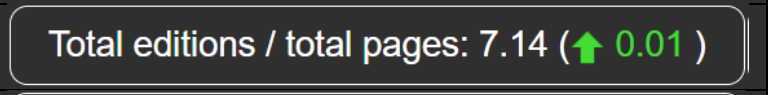
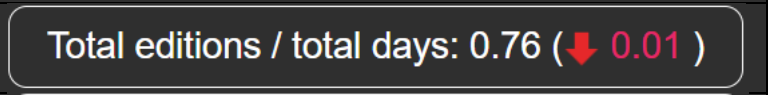


Figura 31. Ejemplo de pestaña de estadísticas

La barra de estado da una perspectiva general del estado de la wiki en ese momento, puesto que muestra magnitudes que son acumuladas, es decir, tiene en cuenta toda la actividad de la wiki hasta el momento. La información se representa mediante iconos significativos, mientras que los datos derivados tienen etiquetas más explicitas. La información mostrada es la siguiente:

Número total de páginas	
Número total de ediciones	
Número total de usuarios registrados	
Número total de usuarios anónimos	
Ratio páginas entre usuarios	
Ratio ediciones entre páginas	
Ratio ediciones entre usuarios	
Usuarios activos ese mes	

A continuación, hay una tabla con un ranking global de usuarios, ordenados por el número de ediciones. A su derecha, un diagrama de barras horizontales que representa la repartición de las ediciones del mes entre los usuarios activos. A su vez cada barra está dividida entre las páginas que ha editado ese usuario.

User	Editions	Creations
Lalagunanegra	995	173
Con Carne	227	90
Maryo96	158	89
Thesweetlove3	38	16
Godric Potter	37	2
DJMintOne	26	3
2501	23	0
Axlish Fuegoscuro	18	8
Flyssi	15	0
Oliver0796	14	3

Figura 32. Top Usuarios

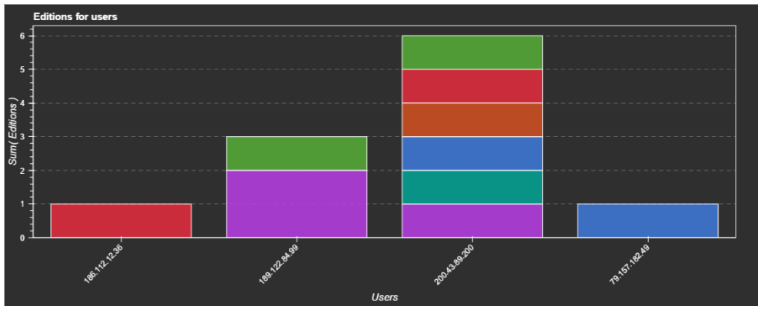


Figura 33. Distribución de las ediciones entre los usuarios

La información de la parte inferior es similar, pero en este caso relativa a las páginas, mostrando su ranking global, así como la repartición de las ediciones del mes entre las páginas editadas, con la subdivisión de los usuarios que las han editado.

ID	Name	Editions	Creation date	Contributors
1	Laguna Negra Wiki	116	8 years ago	19
43	Roque	103	7 years ago	25
7	Marcos	87	8 years ago	27
15	Max Levov	87	8 years ago	12
32	Iván	77	7 years ago	24
181	Proyecto Géminis	65	7 years ago	30
10	Carolina	63	8 years ago	21
42	Plantilla:Infobox Personaje	59	7 years ago	6
38	Ritter Wulf	53	7 years ago	18
80	Fermín	51	7 years ago	15

Figura 34. Top Páginas

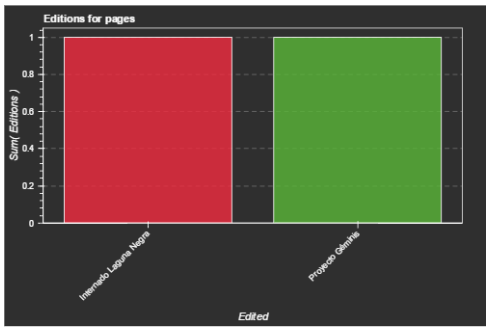


Figura 35. Distribución de las ediciones entre las páginas

9. EJEMPLO DE USO

A continuación, se procederá al análisis de una wiki de prueba, para poder ver qué tipo de información se puede analizar. La wiki elegida ha sido la wiki de Laguna Negra. En primer lugar, observamos las magnitudes más básicas, es decir, número total de páginas y número total de ediciones, y podemos ver que tienen una forma similar. Esto nos puede llevar a pensar que las ediciones por página son más o menos estables

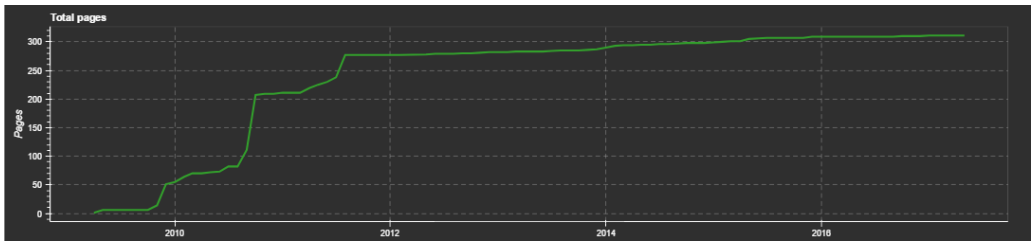


Figura 36. Evolución de páginas totales de Laguna Negra wiki

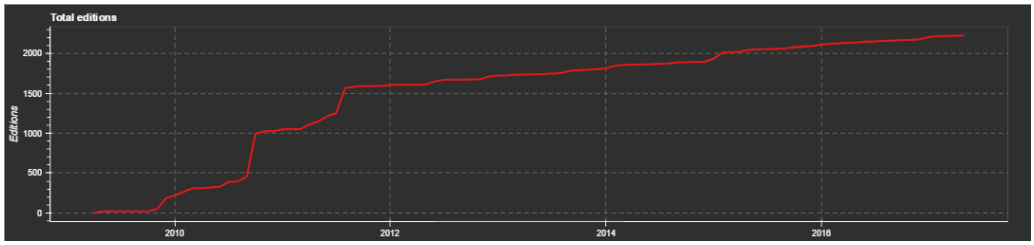


Figura 37. Evolución del número total de ediciones de Lguna Negra wiki

Esa magnitud está representada en una gráfica, y podemos comprobar que casi desde el principio, este cociente se mantiene entre las 5 y las 7 ediciones por página en los últimos 6 años.

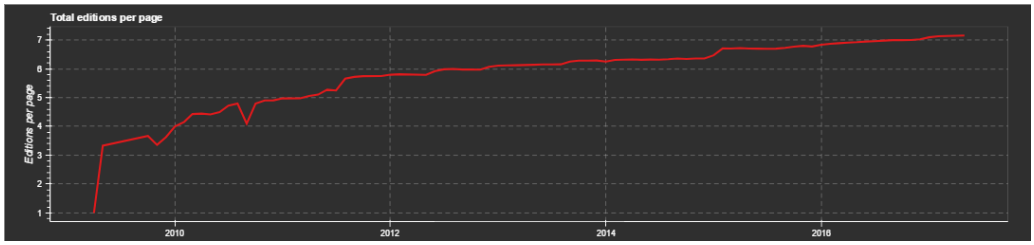


Figura 38. Evolución del promedio de ediciones por página de Laguna Negra wiki

También habría que mirar el comportamiento de los usuarios, que en general suele estar muy relacionado con las ediciones. En particular esta wiki es un caso muy raro, porque los usuarios anónimos representan la gran mayoría de usuarios. Podemos observar en las siguientes figuras, cómo han ido evolucionando las altas de usuarios

de ambos tipos, con lo que vemos que apenas ha habido altas de usuarios registrados desde su nacimiento.

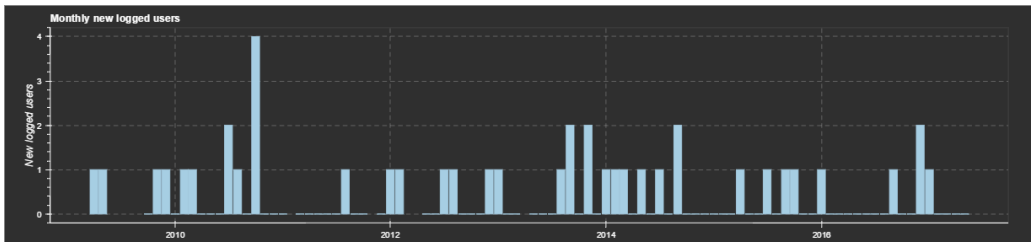


Figura 39. Nuevos usuarios registrados de Laguna Negra wiki

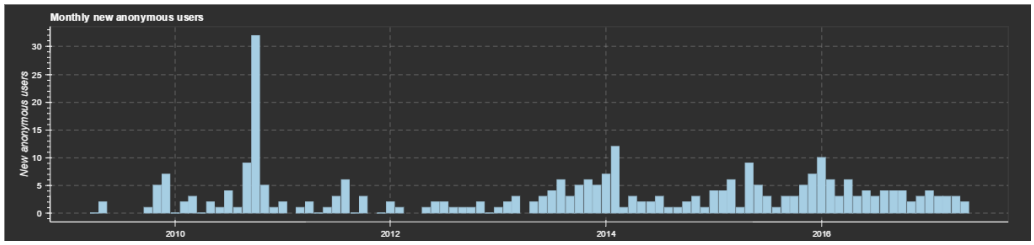


Figura 40. Nuevos usuarios anónimos de Laguna Negra wiki

Esta situación se puede comprobar en la pestaña de estadísticas, donde podemos apreciar que en el momento en que se hizo la exportación de la base de datos, el número de usuarios anónimos superaba en casi 8 veces el número de usuarios registrados.

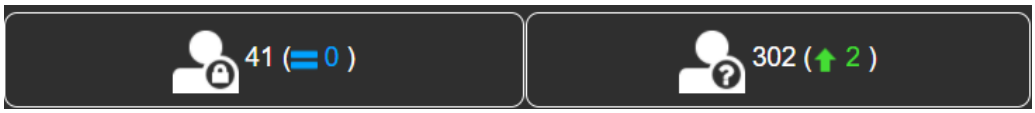


Figura 41. Usuarios registrados vs usuarios anónimos de Laguna Negra wiki

Aunque no se aprecian bien los datos concretos, se puede apreciar un crecimiento importante tanto de las ediciones como de las páginas alrededor de octubre de 2010. Al fijarnos en las mismas fechas de las gráficas de usuarios podemos ver que esos picos también existen, por lo que no es un aumento de actividad esporádica de los usuarios ya existentes, sino que se debe a un aumento en el número de usuarios.

Estos son sólo algunos de los ejemplos de la información que se puede extraer de una wiki utilizando la herramienta. Las posibilidades son muy amplias, y cada wiki será interesante en distintos aspectos. Lo importante es, sobre todo, observar la evolución de las distintas métricas y ver si existe cierta correlación.

10. CONCLUSIONES Y TRABAJO FUTURO

10.1. Conclusiones

El resultado final de este trabajo es una herramienta que se puede utilizar para analizar la evolución de una comunidad colaborativa, como lo es cada una de las que forman el ecosistema de Wikia. Gracias a la procesado de los datos proporcionados por los data-dumps se ha conseguido obtener información valiosa a la hora de buscar causalidades en los cambios de la actividad de la wiki. En particular, gracias a la herramienta, se han podido realizar análisis preliminares que permiten observar cómo cambios en la política de permisos (permitir o no ediciones anónimas) tienen un impacto muy grande en la evolución de una wiki, ya que es un patrón que se observa en múltiples wikis de tamaño medio. Se espera que estos análisis desemboquen en la publicación de un artículo científico sobre análisis de este tipo de comunidades.

Por supuesto, en un entorno como el de Wikia, cada comunidad tiene su propia idiosincrasia y particularidades, distintas de otras. Sin embargo, y con la ayuda de esta herramienta, se puede intentar buscar una correlación o una dependencia entre cosas como las altas de usuario nueva y el crecimiento de páginas, las políticas de ediciones anónimas y el estancamiento de una wiki, o cualquier otra cosa que pueda intentar explicar por qué unas triunfan y otras no de forma cuantitativa.

Además, el dashboard de estadísticas permite hacer un estudio aún más detallado de qué estaba pasando en la wiki en un determinado momento.

10.2. Trabajo futuro

Aún queda mucho por avanzar en este terreno. Este proyecto únicamente ha sido un primer paso a la hora de proporcionar una herramienta de análisis para estas comunidades. A continuación, se desarrollarán algunas propuestas de continuación del trabajo aquí realizado:

- **Plugin de Wikia:** la utilización de esta herramienta requiere que el usuario se descargue el data-dump, instale la herramienta y la despliegue en su ordenador. Sin embargo, puede resultar interesante un plugin que permita monitorizar en tiempo real el estado de la wiki, sin necesidad de descargar la información.
- **Comparación entre wikis:** desafortunadamente, la gran capacidad de cálculo requerida para esta herramienta hace que procesar la información de más de una wiki se haga pesado para el usuario. Puesto que el objetivo es entender mejor las wikis, es de gran interés poder visualizar esta información de distintas wikis en la misma gráfica.
- **Analizar wikis de mayor tamaño:** en el desarrollo de este proyecto se ha detectado que el generador de data-dumps no lo realiza correctamente para wikis de gran tamaño. Para estudiar casos de wikis de éxito sería fundamental conseguir visualizar la información de estas comunidades.

- **Precisar tipos de usuario:** aunque no viene especificado explícitamente en los datos, podría resultar interesante intentar identificar los tipos de usuarios definidos por Wikia en función de la actividad que tienen en la wiki. Si se consigue esto, tampoco sería descartable la utilización de algún mecanismo de aprendizaje automático para intentar predecir el rol de un usuario.
- **Mejora de la interfaz:** en este proyecto la atención se ha centrado en el análisis de datos. Sin embargo, existe un amplio margen de mejora en la interfaz de usuario, y así hacer la experiencia de usuario más atractiva. Esta mejora seguramente conllevaría un uso más extendido de la aplicación, puesto que puede resultar fundamental para el usuario a la hora de decidir entre herramientas.

10. CONCLUSIONS

10.1. Conclusions

The final result of this project is a tool that can be used to analyze collaborative communities like the ones that gathers Wikia. Thanks to the processing of the data provided by the data-dump valuable information has been gathered in order to find the causalities of the changes in a wiki. In particular, with the help of the tool, it has been noticed that changing the policy of anonymous editions has carried great consequences in the evolution of the wiki. This has been observed in multiple mid-sized wikis. A scientific publication is to be expected about the analysis of these type of communities.

Of course, in an environment like Wikia, every community is very different. However, with the help of this tool, the search of correlation between things like new users and the growth of the pages or anonymous edition policies can be done more easily. This key to understanding why some wikis keep growing while others fall into oblivion.

Furthermore, the statistics dashboard allows the user to have a closer look to the activity of a wiki in a certain moment

10.2. Future work

Despite all, there is much more ground to cover in this subject. The project has only been a first step in providing tools for the analysis of these communities. Here is some of the roads that could be followed by those who wish to take this project even further:

- **Wikia plugin:** the use of this tool requires the user to download the data-dump, install the tool and deploy it on his or her computer. However, it can be rather interesting to build a plugin to show the same information but in live timing, without having to go through this perhaps messy process to update the information.
- **Wiki comparison:** unfortunately, the great processing capacity needed for this tool makes using the data of two or more wikis not a pleasant experience for the user. Given that the objective is to understand wikis and therefore compare them, it would be most interesting to visualize the data of more than one wiki at the time.
- **Analyzing bigger wikis:** in the development of this project, it has been noticed that the tool that creates the data-dumps is not working properly when used in a large wiki. In order to analyze some success cases this information would be most helpful.
- **Identifying types of users:** although it is not specified in the data-dump, it would be interesting to deduce the type of the users from its activity. Furthermore, perhaps some machine learning algorithms can help predict the roles of the users.

- **Improving the user interface:** this project has been focused on data analysis, giving less importance to the interface. Therefore, there is room for improvement in this area and this would probably mean more acceptance from potential users. A friendly and attractive user interface can be a key factor when users are choosing between similar tools.

10. BIBLIOGRAPHY

[1] «GNU,» Available: https://www.gnu.org/gnu/linux-and-gnu.es.html.

[2] «Decide Madrid,» Available: https://decide.madrid.es/.

[3] W. C. Bo Leuf, «The Wiki Way: Quick Collaboration on the Web».

[4] C. M. Rodriguez, Wikipedia: Inteligencia colectiva en la red.

[5] D. J. Barrett, MediaWiki: Wikipedia and Beyond.

[6] «Jimmy Wales,» Available: https://es.wikipedia.org/wiki/Jimmy_Wales.

[7] «Larry Sanger,» Available: https://es.wikipedia.org/wiki/Larry_Sanger.

[8] «Estadísticas de Wikipedia,» Available: https://meta.wikimedia.org/wiki/List_of_Wikipedias.

[9] «Wikiversity,» Available: https://es.wikiversity.org/wiki/Portada.

[10] «Ganfyd,» Available: http://www.ganfyd.org/index.php?title=Main_Page.

[11] D. Laura, «Social coding in GitHub: transparency and collaboration in an open software repository,» 2012.

[12] B. Vasilescu, «Perceptions of Diversity on Git Hub: A User Survey,» 2015.

[13] B. M. H. &. A. Shaw, «Laboratories of Oligarchy? How The Iron Law Extends to Peer Production».

[14] F. Ortega, «The Rise and Fall of an Online Project. Is Bureaucracy Killing Efficiency in Open Knowledge Production?».

[15] Y. Benkler, The Wealth of Networks: How Social Production Transforms Markets and Freedom, 2006.

[16] «Uber,» Available: https://www.uber.com/es-ES/.

[17] «Airbnb,» Available: https://www.airbnb.es/.

[18] J. a. H. M. Kolbitsch, «The Transformation of the Web: How Emergin Communities Shape the Information We Consume,» 2006.

[19] J. Nielsen, «The 90-9-1 Rule for Participation Inequality in Social Media and Online Communities,» 2006.

[20] J. Freeman, «The Tiranny of Structurelessness».

[21] R. Michels, Political Parties, 1911.

[22] «Bitergia,» Available: https://bitergia.com/.

[23] «Edgesense,» Available: http://catalyst-fp7.eu/open-tools/edgesense/.

[24] «Drupal,» Available: https://www.drupal.org/.

[25] «Wikitalia,» Available: http://www.wikitalia.org/.

[26] «ChartsUp,» Available: http://chartsup.esy.es/.

[27] A. P. &. A. V. Claudia Gil, «Visualización de datos de comunidades colaborativas,» *Universidad Complutense de Madrid,* 2016.

[28] K. Welsh, «http://www.wolffolins.com/views/32328084745/infograph-epoch,»

[29] S. Data, «seeingdata.org,» Available: http://seeingdata.org/.

[30] S. Data. Available: http://seeingdata.org/developing-visualisation-literacy/key-terms-in-visualisation/.

[31] «Andy Kirk,» Available: http://www.visualisingdata.com/about/.

[32] «Visualizing Data,» Available: http://www.visualisingdata.com/.

[33] H. Kennedy. Available: http://blogs.lse.ac.uk/impactofsocialsciences/2015/07/22/seeing-data-how-people-engage-with-data-visualisations/.

[34] «MIT License,» Available: https://opensource.org/licenses/MIT.

[35] «Python,» Available: https://www.python.org/.

[36] «Guido van Rossum,» Available: https://en.wikipedia.org/wiki/Guido_van_Rossum.

[37] «Open Source Initiative,» Available: https://opensource.org/osd.

[38] «R project,» Available: https://www.r-project.org/.

[39] «Matlab,» Available: https://www.mathworks.com/products/matlab.html.

[40] «NumPy,» Available: http://www.numpy.org/.

[41] «Sublime Text 3,» Available: https://www.sublimetext.com/.

[42] «NumPy,» Available: http://www.numpy.org/.

[43] «SQLite,» Available: https://www.sqlite.org/.

[44] «SQLite3 Python,» Available: https://docs.python.org/3/library/sqlite3.html.

[45] «SQL,» Available: https://en.wikipedia.org/wiki/SQL.

[46] «Bokeh,» Available: http://bokehplots.com/pages/about-bokeh.html.

[47] «BokehJS,» Available: http://bokeh.pydata.org/en/latest/docs/dev_guide/bokehjs.html.

[48] «Bokeh Serve,» Available: http://bokeh.pydata.org/en/latest/docs/user_guide/server.html.

[49] «CSS,» Available: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo.

[50] «Github,» Available: http://conociendogithub.readthedocs.io/en/latest/data/introduccion/.

[51] «Git,» Available: https://es.wikipedia.org/wiki/Git.

[52] «GRASIA,» Available: https://grasia.fdi.ucm.es/main/.

[53] «Repositorio del grupo de investigación GRASIA,» Available: https://github.com/GRASIA.

[54] «Descarga del data-dump,» Available: http://comunidad.wikia.com/wiki/Ayuda:Descargar_la_base_de_datos.

[55] «Untangle,» Available: https://untangle.readthedocs.io/en/latest/.

[56] «XML Element Tree,» Available: https://docs.python.org/2/library/xml.etree.elementtree.html.

[57] «Descarga de Anaconda,» Available: https://www.continuum.io/downloads.

[58] «JavaScript,» Available: https://www.javascript.com/.

[60] «Licencia de Python,» Available: https://docs.python.org/3/license.html.

[61] «Licencia GPL,» Available: https://www.gnu.org/licenses/gpl-3.0.en.html.

[62] «Licencia BSD,» Available: https://es.wikipedia.org/wiki/Licencia_BSD.

ANEXO: MANUAL DE INSTALACIÓN

Prerrequisitos de software

Para utilizar esta aplicación es necesario tener instalados lo siguiente:

- Python3, al menos la versión 3.5.
- Bokeh 0.12.5.
- SQLite 3.8.6 (esta es la versión con la que se ha probado, puede funcionar con otras, pero no se garantiza).

Para facilitar la instalación de todos estos paquetes se puede instalar un software llamado Anaconda [57]. Este software incluye todos los paquetes a utilizar, además de Python. Por defecto, a día de hoy, la instalación no incluye la versión más reciente. Por ello es necesario actualizarlos, lo que se hace con el comando "conda update <paquete>" donde paquete es aquel que se desea actualizar. Por ejemplo:

```
>>conda update bokeh
```

Una vez instalados y actualizados todos los paquetes se puede proceder a la descarga del código.

Ejecución del código

Es necesario guardar el data-dump (descargado como se explica en la sección en la carpeta dashboard explicada que en la sección de la arquitectura Para ejecutar la aplicación es necesario viajar en la línea de comandos hasta la carpeta padre de la carpeta dashboard. Una vez ahí se puede empezar a ejecutar. Los pasos explicados a continuación son para el sistema operativo Windows. Los detalles para sistemas Linux se pueden encontrar en el Readme del repositorio GitHub.

El primer paso es parsear el data-dump a un fichero de texto. Esto se hace con el siguiente comando:

```
>>python dump_parser.py <file.xml>
```

Donde file.xml es el dump descargado y descomprimido. A continuación, se realiza la carga en la base de datos:

```
>>python dataHandler.py <file.txt>
```

Donde file.txt es el resultado del paso anterior. Por último, la aplicación ya estaría lista para ser desplegada, lo que se hace de la siguiente manera:

```
>>bokeh serve --show dashboard
```

Esto crearía un proceso en el puerto 5006 de localhost. La opción show hace que se nos abra automáticamente el navegador. La URL donde está la aplicación es:

http://localhost:5006/dashboard