Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
(Mención Computación)

Extracción de titulares de periódicos on-line a partir de RSS para su posterior estudio

Autor:
D. Adrián Poncela Gómez

Tutor:
D. Quiliano Isaac Moro

Resumen:

Este Trabajo de Fin de Grado tiene como objetivo el desarrollo de una aplicación de extracción y limpieza de titulares de prensa online, mediante el RSS para un posterior estudio estadístico.

El usuario a través de una interfaz, podrá seleccionar en un menú los datos que quiere recoger, así como el tiempo de ejecución del programa. La herramienta mostrará unos resultados en tiempo real con gráficos de cómo va el proceso de obtención de los titulares.

Abstract:

The objective of this Final Degree Project is the development of an online press headline extraction and cleaning application, using RSS for a subsequent statistical study.

The user, through an interface, will be able to select from a menu the data he wants to collect, as well as the execution time of the program. The tool will show results with graphs of how the process of obtaining the headlines is going.

Índice general

Índice de figuras

Índice de tablas

1 Contextualización y objetivos

1.1 Contextualización

Actualmente, nuestra sociedad se encuentra en la denominada "Era de la Información" (El País, 2016), también conocida como "Era Digital" o "Era Informática". Así se denomina al periodo de la historia de la humanidad que va ligado a las tecnologías de la información y la comunicación (TIC). El comienzo de este periodo se asocia con la revolución digital.

En este mundo altamente globalizado, en el que vivimos, que se encuentra sujeto a los cambios que las nuevas tecnologías propician, es una realidad que desde hace un tiempo el periodismo está sufriendo un gran cambio, adaptándose a las nuevas tecnologías de la información.

El periodismo digital lleva poco más de una década en el panorama de los medios de comunicación, pero en poco tiempo ha sido capaz de hacer frente a su mayor competidor, el gigante de la prensa escrita en papel, el cuál gozaba de una audiencia y difusión más que considerable.

En nuestros días, es una realidad que el periodismo digital ha conseguido consolidarse satisfactoriamente en el panorama comunicativo como un medio más, de referencia y de enorme difusión.

Desde que los diarios de prensa, crearon su edición digital y surgió un sinfín de publicaciones digitales, muchos son los lectores que han migrado de un medio a otro, ya que el medio digital proporciona una serie de ventajas potenciales que le caracterizan y constituyen su esencia.

Se podría destacar entre sus mayores ventajas y características del nuevo paradigma digital la interactividad entre el emisor y receptor, así como la instantaneidad.

Internet, en los últimos años ha tenido una gran repercusión y un fuerte crecimiento, esto es lo que ha provocado el gran auge de todos los diarios y publicaciones digitales, unido al protagonismo y aumento creciente de las redes sociales, que combinados constituyen un poderoso elemento mediático.

1.2 Objetivos

El objetivo de este trabajo es realizar una aplicación, que pueda satisfacer la necesidad de un usuario de conocer estadísticas de titulares de prensa, y tener la posibilidad de realizar un estudio comparativo entre esos titulares.

Básicamente podríamos decir que consiste en un experimento, en el que el usuario en primer lugar, selecciona los periódicos que desea, así como el tipo de Información que quiere (Deportes, España, Internacional…). El usuario determinará el número de titulares que desea almacenar o el tiempo en el que quiere realizar la extracción, para posteriormente poder realizar el estudio estadístico.

La herramienta extraerá los titulares tal cual han sido publicados y los almacenará en una base de datos. Posteriormente se realizado un tratamiento de esos titulares, que consiste en eliminar aquellas palabras que podemos considerar que tienen poca importancia o relevancia para nuestro estudio estadístico. Estas palabras son los artículos, preposiciones, conjunciones, interjecciones entre otras. Finalmente, cuando el experimento finalice se mostrarán unos gráficos con el que el usuario podrá realizar un análisis estadístico muy elemental de los resultados.

A medida que se van almacenando los titulares, el usuario verá el porcentaje de finalización del experimento, así como los últimos titulares almacenados.

Conviene aclarar que este trabajo no tiene como objetivo la realización de un estudio complejo y detallado de la comparación de los titulares aparecidos en los servicios de internet que ofrecen algunos periódicos. Más bien

tiene como objetivo el crear una plataforma sobre la cual un estadista con conocimientos de manejo de bases de datos pueda analizar dichos datos de una manera más profunda.

Se puede dividir el trabajo en bloques que serían los siguientes:

1) Extracción de titulares,
2) Limpieza.
3) Almacenamiento.
4) Visualización Estadísticas

El bloque más importante quizás es el primero. Para la extracción de los titulares, utilizaremos el RSS (Wikipedia, 2019) que nos proporcionan las páginas de periódicos online, y una lectura de un fichero XML.

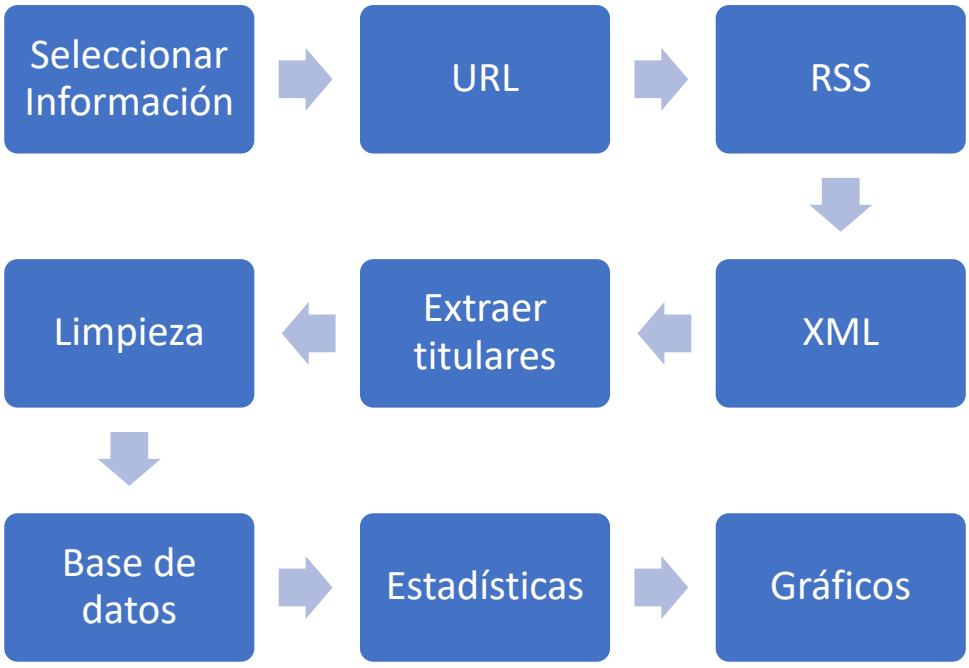
El segundo bloque, se realizará mediante una lista formada con las palabras consideradas menos relevantes, que se desean eliminar. El resultado de esta limpieza, será un titular formado por palabras que principalmente serán verbos, sustantivos, nombres propios y adjetivos. Estas palabras sí se pueden considerar que tienen la importancia suficiente dentro del titular.

El almacenamiento consistirá en guardar no sólo el titular, sino el titular tras el proceso de limpieza, así como el periódico al que pertenece y el tipo de información que trata. A mayores se incluirá la fecha de publicación de dicho titular con unas estadísticas básicas como son el número de palabras y el número de caracteres que contienen.

También se guardarán en una tabla todas las palabras que no han sido eliminadas en el proceso de limpieza, es decir, las que se consideran con la suficiente relevancia. Indicando a qué titular pertenece, y por tanto en qué fuente y tipo de información ha aparecido.

Por último, la visualización de estadísticas consistirá en la representación de unos gráficos básicos creados a partir de la información almacenada. Además, se enseñará un ejemplo de utilización de un programa que permite una gran posibilidad de representaciones distintas y avanzadas que un usuario podría utilizar para realizar un estudio estadístico más profundo.

1.3 Esquema general



1.4 Tecnología empleada

Para el desarrollo de este TFG se han utilizado las siguientes herramientas, así como lenguajes de programación y programas:

- Lenguaje Java
- NetBeans
- Java Swing
- MySQL
- MySQL WorkBench
- Astah UML
- Git
- Microsoft Project
- Lenguaje R
- RStudio
- Power Bi Desktop

En el Capítulo 4, que trata sobre las "Herramientas Utilizadas" se explicarán con más detalle qué son y el por qué de su utilización.

1.5 Metodología utilizada

Tras hacer una valoración de las distintas posibilidades de planificación existentes como pueden ser alguna metodología ágil, el método en cascada o el proceso unificado. Para la realización de este TFG se ha utilizado una metodología que se basa en un desarrollo iterativo e incremental como es el Proceso Unificado.

En el Capítulo 2 que trata sobre la "Planificación" se explicará con más detalle en qué consiste esta metodología y cómo se ha utilizado.

1.6 Estructura de la memoria

Esta memoria del TFG está estructurada por capítulos. A continuación, se explica brevemente qué contenidos se tratan en cada uno de ellos, así como las secciones que contienen.

- Capítulo 1. Introducción: En este apartado se mostrará una contextualización para este TFG, así como una breve exposición de los objetivos, la metodología utilizada y el contenido del CD.

- Capítulo 2. Planificación: Se incluirá la explicación de la metodología utilizada, así como su justificación. Se expondrán la gestión de riesgos del TFG, así como las tareas realizadas a lo largo del tiempo. Por último, también se incluirá el coste de este proyecto.

- Capítulo 3. Análisis y especificación de requisitos: En este apartado se mostrarán los requisitos funcionales, los requisitos no funcionales y los requisitos de información. Además, se mostrarán los casos de usos, actores, modelo de dominio y diagramas de secuencia.

- Capítulo 4. Herramientas utilizadas: Se explicarán las diferentes herramientas utilizadas, así como los lenguajes de programación y programas utilizados. También una justificación de la decisión de su utilización.

- Capítulo 5. Diseño: Explicación de los patrones de diseño y arquitectónicos que han sido utilizados en este TFG.

- Capítulo 6. Implementación: Se mostrarán cómo se ha desarrollado la herramienta en cuanto a código se refiere, así como su justificación de por qué se ha realizado de ese modo.

- Capítulo 7. Pruebas: Pruebas realizadas durante el desarrollo de la herramienta implementada.

- Capítulo 8. Conclusiones: En este último capítulo se expondrán los objetivos alcanzados, valoración personal del alumno. También se comentarán las posibles mejoras de este TFG, en lo que sería un trabajo futuro.

- Bibliografía: Documentación utilizada y consultada a lo largo de la programación de la herramienta cómo en la realización de la memoria de este TFG.

- Anexos: Incluye el manual de usuario, manual de instalación y mantenimiento, así como anexos de código.

1.7 Contenido del CD

En el interior del CD se puede encontrar:

- Documento con la memoria en PDF.
- Carpeta con el proyecto exportado de NetBeans.
- Carpeta con las bibliotecas necesarias para incluir en el proyecto, en formato JAR.
- Carpeta con ejemplos de archivos de palabras negativas, positivas y de búsqueda.
- Script SQL.

2 Planificación

En este capítulo se explicará la planificación de este TFG. La planificación es una parte fundamental en el desarrollo de un proyecto. Conviene planificar y determinar en qué orden se van a realizar las tareas, así como la estimación de tiempo que va a suponer la realización de cada una de ellas.

Es necesario realizar esta planificación, antes de comenzar con el análisis, el diseño y la implementación de la herramienta.

Se explicará la metodología utilizada, la gestión de riesgos, la gestión de recursos y por último se realizará un cálculo sobre el coste del proyecto.

Uso de Proceso Unificado
Lo primero de todo en cuanto a la planificación del proyecto se refiere es qué tipo de metodología se va a seguir.

En este punto se pueden analizar algunas metodologías y hacer una valoración de cuál se puede adaptar mejor. Se valoran la utilización de una metodología ágil, cascada o proceso unificado.

Se descarta la utilización de un modelo de desarrollo en cascada debido a la más que posible necesidad de modificar los requisitos a medida que se avanza en el proyecto.

La utilización de una metodología ágil cómo podría ser SCRUM, también se descarta, debido a que es necesario un equipo de desarrollo con experiencia, no siendo el caso de este TFG.

Descartando el modelo de desarrollo en cascada y el uso de una metodología ágil, se decide utilizar el proceso unificado. La razón de esta decisión, se encuentra en que este método proporciona un enfoque iterativo e incremental. A diferencia de la metodología ágil no requiere personal con experiencia con este tipo de planificación.

2.1 Proceso Unificado de desarrollo de software

En este apartado se procede a explicar en qué consiste el Proceso unificado (Wikipedia, 2019) de desarrollo de software.

El Proceso Unificado es el marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, estar centrado en la arquitectura y ser iterativo e incremental.

2.1.1 Características

- Iterativo e incremental

  El Proceso Unificado está formado por cuatro fases que se denominan: Inicio, Elaboración, Construcción y Transición. Cada uno de las fases, se divide en una serie de iteraciones.
  Las iteraciones tienen como finalidad, mostrar un incremento del producto desarrollado añadiendo una nueva funcionalidad o mejorando las funcionalidades ya existentes.

- Dirigido por casos de uso

  En el Proceso Unificado los casos de uso son utilizados para establecer los requisitos funcionales. También se utilizan para definir los contenidos de cada una de las iteraciones.
  La idea es que cada iteración tome un conjunto de casos de uso y se desarrolle todo el proceso a través de las distintas disciplinas como pueden son la implementación, diseño y las pruebas.

- Centrado en la arquitectura

  Con el Proceso Unificado se asume que no hay un único modelo con el que se cubran todos los contenidos del sistema. Debido a este motivo existen varios modelos y vistas con la que se define la arquitectura de software de un sistema.

  Se parte de una visión global del sistema que se va refinando poco a poco hacia niveles inferiores para poder definir así cada uno de los componentes básicos que formarán el sistema.

- Enfocado en los riesgos

  El Proceso Unificado necesita que el equipo de proyecto consiga identificar cada uno de los riesgos críticos en una etapa temprana del ciclo de vida.

  Se realiza con la finalidad de disminuir la probabilidad de que un riesgo ocurra, siendo estos tratados con la mayor prioridad.

2.1.2 Fases

El Proceso Unificado se divide en cuatro fases:

- Inicio:

  La fase de inicia tiene su importancia principalmente en nuevos desarrollos donde hay importantes riesgos que pueden ocurrir. A mayores existen una serie de requisitos que deben ser abordados antes de que el proyecto pueda continuar.

  Cuando la finalidad es una mejora de lo que ya existía, se trata de una fase más breve, centrándose principalmente en la viabilidad del proyecto.

  Objetivos y tareas importantes de esta fase:
  - Descripción del producto final.
  - Presentar análisis de negocio.
  - Identificar por mayores riesgos potenciales.
  - Establecimiento de las funcionalidades del sistema.
  - Identificar los casos de uso más importantes y dependientes.
  - Estimación del coste y presupuesto.
  - Cronograma de tareas.

  Con el hito de los objetivos de desarrollo, finaliza la fase de inicio.

- Elaboración:

  En esta fase se obtiene una visión refinada de lo que va a resultar el proyecto. Se realiza la implementación del núcleo de la herramienta. Se resuelven aquellos riesgos más importantes que se pueden considerar críticos. Se pueden añadir nuevos requisitos. También se intentan ajustar las estimaciones.

  La mayor parte de los requisitos funcionales del sistema tienen que ser capturados en esta fase, además se tiene que diseñar la arquitectura inicial del programa.

  Todo esto tiene la finalidad de proporcionar una base estable, de cara afrontar las fases que requieren un mayor esfuerzo.

  Con el hito de la arquitectura del sistema, se da por finalizada la fase elaboración.

- Construcción:

  Se trata de la fase más extensa dentro del desarrollo.

Se parte de la base de la arquitectura obtenida de la fase elaboración, con ella, en cada iteración se trata de evolucionarla hasta convertirse en un producto listo incluyendo requisitos mínimos. Con la finalización de cada iteración, se consigue una versión nueva ejecutable del producto, con funcionalidades añadidas o mejoradas, con respecto a las de la iteración anterior.

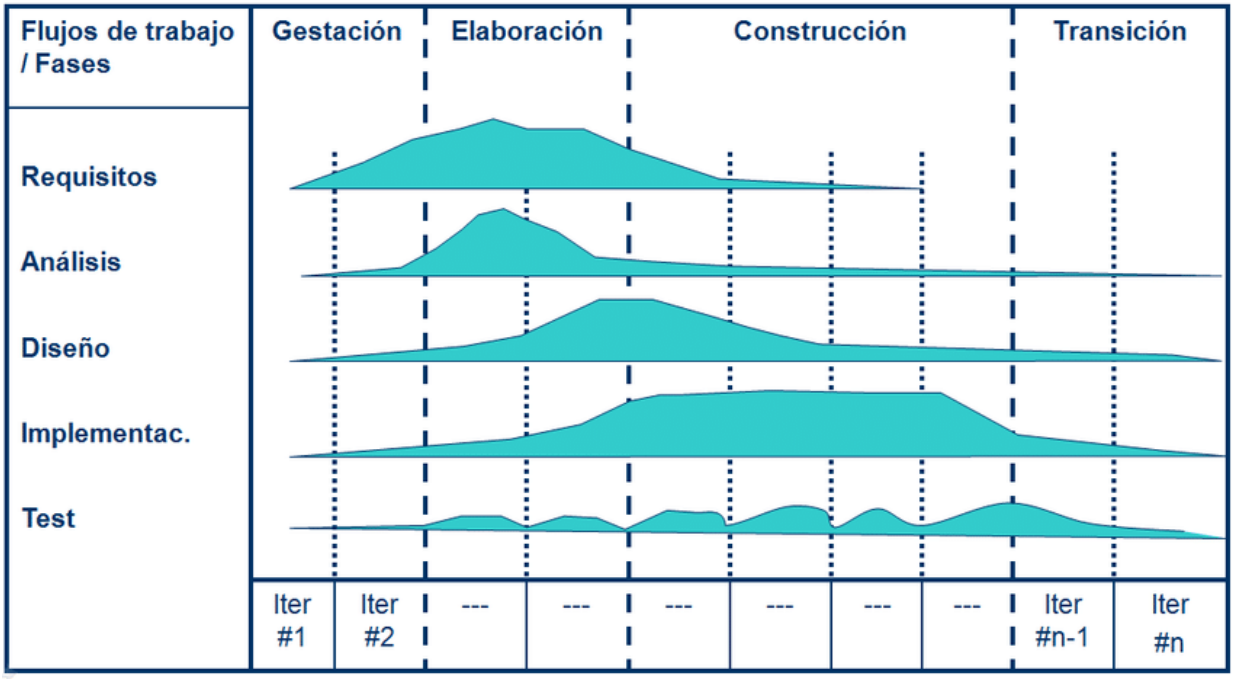
Se tratan los riesgos de menor importancia

Con el hito de la obtención de una aplicación con todas las funcionalidades, se da por completa la fase de construcción.

- Transición:

Se trata de la última fase, en la que el producto debe de estar preparado para ser probado y utilizado por el cliente sin problemas. Tras esto, se pensará en aumentar las funcionalidades para la mejora del producto.

Durante esta fase se debe llevar a cabo la validación del sistema, así como resolver todos aquellos falos o errores identificados.
La fase concluye con el producto final.



(Researchgate, 2019)

Figura 1. Proceso Unificado

2.2 Gestión de Riesgos

Una de las partes más importantes a la hora de planificar el desarrollo de un proyecto es la gestión de los riesgos. No tener en cuenta los riesgos o darles menos importancia de la que realmente tienen, puede suponer un fracaso en el proyecto, así como el no cumplimiento de los objetivos establecidos.

En este apartado se expondrán los riesgos que han sido detectados, acompañados de una descripción de los mismos, incluyendo su impacto y probabilidad. Junto con la matriz de exposición que se adjunta, se determina para cada uno de los riesgos, la exposición existente a dicho riesgo.

Lo primero de todo es determinar que se entiende por un riesgo: El PM-BOK proporciona la siguiente definición: “Un riesgo es un evento o una condición incierta que, si ocurren, tienen un efecto positivo o negativo sobre los objetivos del proyecto”

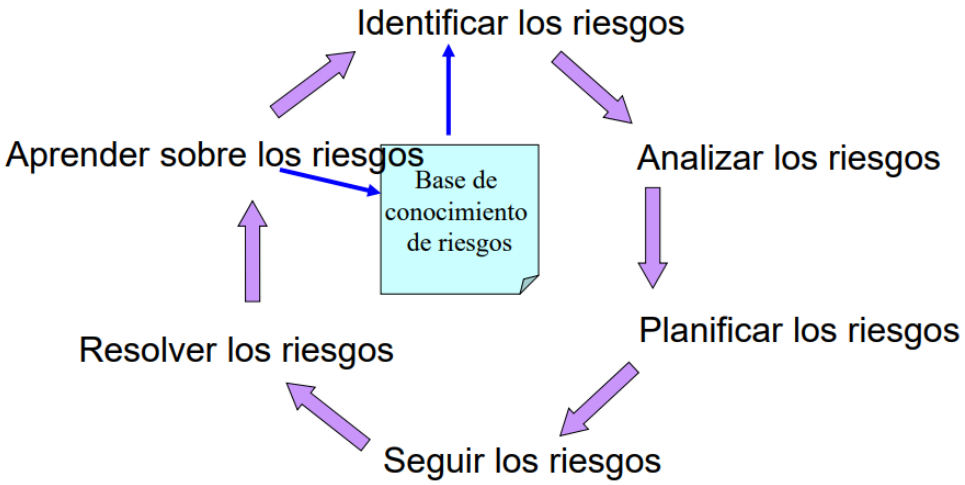
Por tanto, el plan de gestión de riesgos tiene la finalidad de controlar, valorar y prevenir los principales riesgos que pueden ocurrir, afectando adversamente al proyecto. También incluye, para cada uno de los riegos un plan de acción para evitar o minimizar el impacto que pueda suponer.

Etapas que componen el plan de riesgos:

- Identificación de todos aquellos riesgos que puedan aparecer durante el desarrollo del proyecto, en cualquier etapa.
- Análisis de cada uno de los riesgos que han sido identificados, asignándolos una probabilidad de ocurrencia, nivel de impacto. Se clasificarán en función de a qué parte o etapa del proyecto afectan.

- Creación de un plan de acción para cada riesgo, aunque especialmente, para aquellos que sean más importantes, indicando que decisiones o acciones se debe tomar cuando sucedan.

- Gestionar y supervisar los riesgos que afectaron al proyecto.



(PDSC, 2019)

Figura 2. Gestión de riesgos

Hay que tener en cuenta, que durante la planificación inicial del proyecto se prevén una serie de riesgos y a medida que se avanza con el desarrollo del proyecto pueden ocurrir esos riesgos previstos, pero también pueden suceder algunos que no han sido valorados. Por tanto, en este plan de gestión de riesgos se incluirán, también todos aquellos riesgos descubiertos durante el desarrollo del proyecto.

Esto se debe a la característica del Proceso Unificado, de en cada iteración revisar y actualizar la gestión de riesgos durante el desarrollo del proyecto.

Se procederá a realizar una clasificación de los riesgos en tres categorías atendiendo a los siguientes criterios:

- Riesgos de Proyecto: Restricciones de recursos, interfaces externas, relaciones con los proveedores, políticas internas, problemas de coordinación interna del equipo o del grupo, financiación no adecuada.

- Riesgos de Proceso: Proceso software no documentado, falta de revisiones efectivas de colegas, no prevención de defectos, proceso de diseño pobre, gestión pobre de requisitos, planificación ineficaz.

- Riesgos de Producto: Falta de experiencia en el dominio, diseño complejo, interfaces definidas deficientemente, sistemas de legado poco comprendidos, requisitos vagos o incompletos.

Probabilidades consideras en la valoración de los riesgos:

- Muy baja.
- Baja.
- Media.
- Alta.
- Muy alta.

Impactos considerados en la valoración de los riesgos:

- Despreciable.
- Marginal.
- Crítico.
- Catastrófico.

Estrategias para la resolución y manejo de los riesgos:

- Reducción del riesgo: Reduce la probabilidad y/o consecuencia del riesgo después de que ocurra.
- Reserva del riesgo: Utilizar la planificación reservada previamente o la holgura del presupuesto,
- Protección del riesgo: Reduce la probabilidad y/o consecuencia del riesgo antes de que ocurra.
- Mitigación del riesgo: Aceptación de la ocurrencia de un riego, pero reduciendo al máximo su impacto.

A continuación, se muestra la figura con la matriz de exposición utilizada y las tablas con los riesgos considerados en este TFG, así como la estrategia y plan de acción para cada uno de ellos.

Matriz Impacto/Probabilidad

Impacto/ Probabilidad	Muy alto	Alto	Medio	Bajo	Muy bajo
Catastrófico	Alto	Alto	Moderado	Moderado	Bajo
Crítico	Alto	Alto	Moderado	Bajo	Ninguno
Marginal	Moderado	Moderado	Bajo	Ninguno	Ninguno
Despreciable	Moderado	Bajo	Bajo	Ninguno	Ninguno

Figura 3. Matriz impacto/probabilidad

Tabla 1. Riesgo 1

Riesgo 1	Pérdida de datos.
Detalle	Pérdida de todo o de una parte del proyecto. Ya sea código, documentación, base de datos, bibliografía…
Categoría	Riesgo de Proyecto.
Impacto	Crítico.
Probabilidad	Baja.
Exposición	Baja.
Estrategia	Mitigación del riesgo.
Plan de acción	Utilización de gitlab.inf.uva para el control de versiones. (GitLab, 2019)

Tabla 2. Riesgo 2

Riesgo 2	Enfermedad.
Detalle	Incapacidad de poder trabajar en el proyecto temporalmente debido a enfermedad.
Categoría	Riesgo de Proyecto.
Impacto	Marginal.
Probabilidad	Baja.
Exposición	Ninguna.
Estrategia	Mitigación del riesgo.
Plan de acción	Realizar una replanificación del proyecto según sea necesario en función del tiempo perdido por enfermedad.

Tabla 3. Riesgo 3

Riesgo 3	Cambios en los requisitos.
Detalle	Modificación de los requisitos establecidos previamente en el inicio de la planificación del proyecto.
Categoría	Riesgo de producto.
Impacto	Crítico.
Probabilidad	Media.
Exposición	Moderada.
Estrategia	Reducción del riesgo. Reserva del riesgo.
Plan de acción	Uso de más recursos para minimizar las modificaciones producidas. Reserva de holgura de tiempo.

Tabla 4. Riesgo 4

Riesgo 4	Fallo en el diseño.
Detalle	Error a la hora de definir el diseño de toda o una parte de la herramienta.
Categoría	Riesgo de producto.
Impacto	Crítico.
Probabilidad	Media.
Exposición	Moderada.
Estrategia	Protección del riesgo. Reducción del riesgo.
Plan de acción	Corregir el error realizando una replanificación si fuese necesaria.

Tabla 5. Riesgo 5

Riesgo 5	Fallo en la implementación.
Detalle	Error a la hora de implementar toda o una parte de la herramienta.
Categoría	Riesgo del producto.
Impacto	Crítico.
Probabilidad	Media.
Exposición	Moderada.
Estrategia	Protección del riesgo. Reducción del riesgo.
Plan de acción	Corregir el error realizando una replanificación si fuese necesaria.

Tabla 6. Riesgo 6

Riesgo 6	Planificación no realista.
Detalle	Planificación de una herramienta que no va a ser posible ser ejecutada dentro de los límites de tiempo establecidos, por su complejidad.
Categoría	Reserva del riesgo.
Impacto	Crítico.
Probabilidad	Media.
Exposición	Moderada.
Estrategia	Protección del riesgo.
Plan de acción	Replanificación del proyecto ajustándose a los requisitos principales.

Tabla 7. Riesgo 7

Riesgo 7	Falta de conocimiento con herramientas/tecnologías.
Detalle	Tiempo empleado en adquirir los conocimientos necesarios para la utilización de las herramientas/tecnologías necesarias.
Categoría	Riesgo de producto.
Impacto	Marginal.
Probabilidad	Media.
Exposición	Baja.
Estrategia	Reducción del riesgo.
Plan de acción	Uso de tutoriales o preguntar a expertos en dichas herramientas/tecnologías.

Tabla 8. Riesgo 8

Riesgo 8	Recursos no disponibles.
Detalle	Incapacidad de poder desarrollar, diseñar o implementar una parte de la herramienta por falta de algún recurso necesario para tal fin.
Categoría	Riesgo de Proyecto.
Impacto	Marginal.
Probabilidad	Baja.
Exposición	Baja.
Estrategia	Reserva del riesgo.
Plan de acción	Tener alternativas disponibles y preparadas para su utilización.

Tabla 9. Riesgo 9

Riesgo 9	Asuntos personales.
Detalle	Incapacidad de poder trabajar en el proyecto temporalmente debido a asuntos personales.
Categoría	Riesgo de Proyecto.
Impacto	Marginal.
Probabilidad	Baja.
Exposición	Ninguna.
Estrategia	Mitigación del riesgo.
Plan de acción	Realizar una replanificación del proyecto según sea necesario en función del tiempo perdido por asuntos personales.

Tabla 10. Riesgo 10

Riesgo 10	Fallo conexión de Red.
Detalle	La conexión de red se interrumpe en el momento de consultar el archivo.
Categoría	Riesgos de proyecto.
Impacto	Crítico.
Probabilidad	Baja.
Exposición	Baja.
Estrategia	Mitigación del riesgo.
Plan de acción	Si la conexión con alguna página falla, continuar con la siguiente y volver a intentarlo después.

Tabla 11. Riesgo 11

Riesgo 11	Fallo al proporcionar el XML.
Detalle	Fallo en la creación del archivo XML, por parte de la fuente online.
Categoría	Riesgos de proyecto.
Impacto	Crítico.
Probabilidad	Baja.
Exposición	Baja.
Estrategia	Mitigación del riesgo.
Plan de acción	Realizar una comprobación de que los campos son proporcionados correctamente, y si no es así, ignorarlo por el momento y volver a intentarlo después.

Tabla 12. Riesgo12

Riesgo 12	Requisito mal especificado.
Detalle	Especificación errónea de un requisito.
Categoría	Riesgo de producto.
Impacto	Marginal.
Probabilidad	Baja.
Exposición	Ninguna.
Estrategia	Reducción del riesgo.
Plan de acción	Evaluación de los requisitos e identificación de la funcionalidad que supone dentro del producto para su modificación.

Tabla 13. Riesgo 13

Riesgo 13	Mala estimación de las tareas.
Detalle	Estimar menos horas para una determinada tarea de lo que realmente ha costado realizarla.
Categoría	Riesgo de producto.
Impacto	Marginal.
Probabilidad	Alta.
Exposición	Moderada.
Estrategia	Mitigación del riesgo.
Plan de acción	Replanificación del proyecto.

Tabla 14. Riesgo 14

Riesgo 14	Demasiados errores en la etapa de pruebas.
Detalle	Aparición de más errores en el producto de lo previsto durante la fase de pruebas.
Categoría	Riesgo de proyecto.
Impacto	Marginal.
Probabilidad	Media.
Exposición	Baja.
Estrategia	Mitigación del riesgo.
Plan de acción	Aumento de recursos para solventar esos errores lo antes posible.

2.3 Gestión de Recursos

A lo largo del desarrollo de un proyecto se precisan distintos tipos de recursos. La forma de reservar los recursos puede implicar la existencia de restricciones sobre las tareas programadas y por lo tanto incidir en la planificación temporal considerada. Por tanto, una de las tareas del responsable del proyecto será buscar la concordancia entre las tareas planificadas y los recursos disponibles en cada momento.

Hughes y Cotterell (Cotterell, 2002) dividen los recursos en siete categorías:

- **Trabajo**. Miembros del equipo.
  Una persona, que realizará el proyecto, con los roles de analista, desarrollador y tester del producto.

- **Equipamiento**. Material informático.
  Ordenadores disponibles para la realización del producto, tanto en posesión del alumno como facilitados por la facultad.

- **Materiales**. Consumibles de informática, papel, etc.

- **Espacio**. Si se está en una organización existente el espacio ya está disponible, pero si hay que contratar personal adicional hay que contar con ello.
  Bibliotecas o domicilio del desarrollador

- **Servicios**. Algunos proyectos necesitan la contratación de servicios especiales.

- **Tiempo**. Es uno de los recursos principales de los proyectos ya que, a veces, está preestablecido.

- **Dinero**. Es un recurso secundario, se utiliza para comprar otros recursos que serán consumidos o utilizados.

En concreto para este TFG:

- **Trabajo**. Una persona, que realizará el proyecto, con los roles de analista, desarrollador y tester del producto.

- **Equipamiento**. Ordenadores disponibles para la realización del producto, tanto en posesión del alumno como facilitados por la Escuela.

- **Materiales**. Consumibles de informática, papel, etc.

- **Espacio**. Bibliotecas o domicilio del desarrollador

- **Servicios**. No se ha requerido ningún servicio especial.

- **Tiempo**. Fecha límite propuesta para la entrega de este TFG en Julio de 2019.

- **Dinero**. No se ha invertido dinero en ningún programa o tecnología, pero sí hay un coste de material y horas de trabajo. Se verá en el siguiente apartado.

2.4 Planificación inicial

La planificación inicial de este TFG se ha realizado teniendo en cuenta que el tiempo de trabajo esperado para el mismo es de 300 horas.

Se puede hacer una división por bloques del trabajo a realizar. Para cada apartado se hace una estimación en cuanto al número de horas para su realización:

1) Análisis: 20 horas.
2) Elección de las herramientas: 10 horas.
3) Extracción de titulares: 70 horas.
4) Limpieza: 30 horas.
5) Almacenamiento: 60 horas.
6) Visualización Estadísticas: 20 horas.
7) Pruebas. 40 horas.
8) Redacción de la memoria. 50 horas.

2.5 Diagrama

Se muestra el diagrama de Gantt con la planificación inicial, acorde a la estimación de 300 horas para la realización de este TFG.

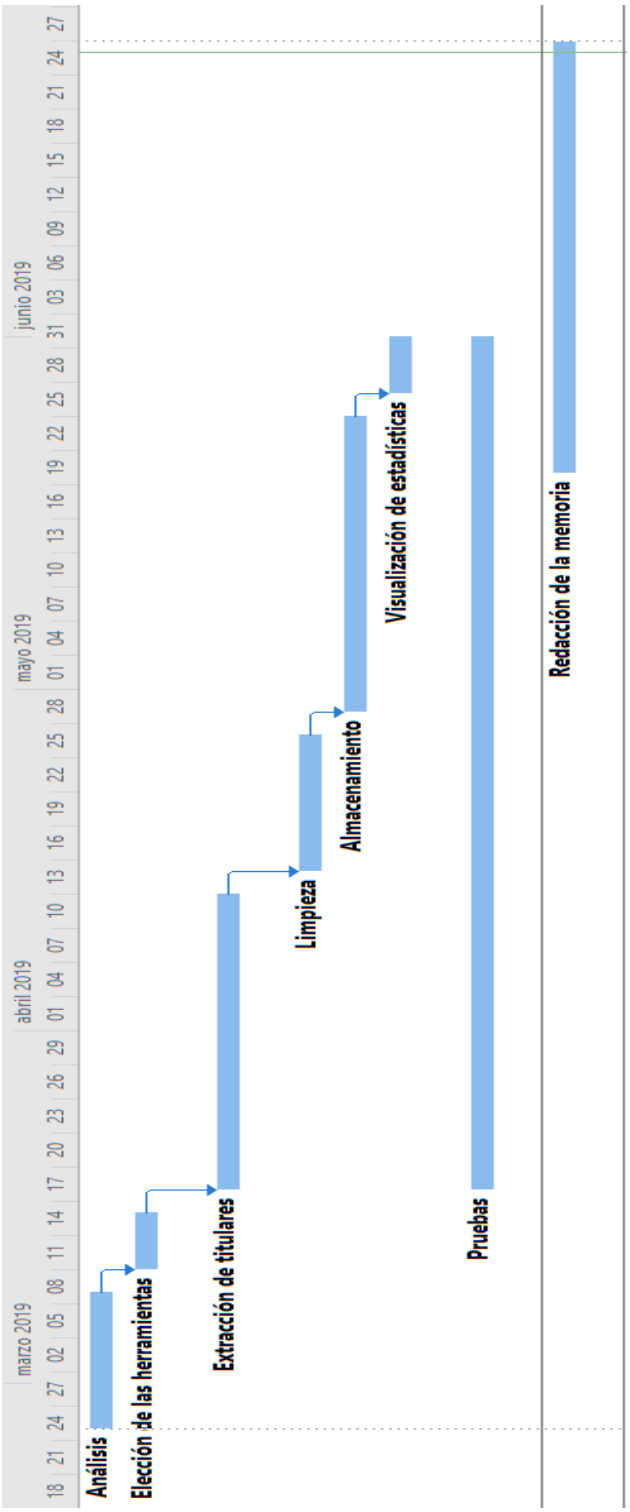


Figura 4. Diagrama Gantt Planificación Inicial

Las pruebas se realizarán periódicamente, al añadir una nueva funcionalidad o modificar alguna ya existente, durante el desarrollo del código.

2.6 Coste del proyecto

En este apartado, se realizará un cálculo de los costes que ha supuesto este TFG. Para la realización de este cálculo se tienen en cuenta tres aspectos: costes de hardware, costes de software y costes de personal.

En cuanto al coste de hardware, se han utilizado 2 ordenadores portátiles.
La decisión de utilizar dos ordenadores se debe a la necesidad de realizar pruebas del programa durante un tiempo de varias horas. Esto sería imposible con un sólo ordenador puesto que necesitaba el otro para realizar las prácticas de empresa. De esta forma, periódicamente realizaba pruebas de extracción de titulares, mientras yo me encontraba trabajando.
A mayores, el programa de visualización de gráficos Power BI, no tiene soporte para MacOS, por lo que necesitaba un ordenador con Windows.

1) MacBook Pro con macOS High Sierra
   Coste aproximado: 800 euros

2) Lenovo Yoga 500 con Windows 8
   Coste aproximado: 350 euros

Teniendo en cuenta una vida media para cada uno de ellos de 5 años y que la duración del TFG ha sido de 4 meses, el coste para cada uno de ellos es el siguiente:

1) (800€ / (5 x 12) meses) x 4 meses = 53€
2) (350€ / (5 x 12) meses) x 4 meses = 23 €

La suma de ambas cantidades, resulta un valor de 76 euros, que serán considerados costes de hardware.

En cuanto a costes de software, hay que tener en cuenta que los sistemas operativos de ambos ordenadores venían incluidos por lo que no ha sido necesario realizar ninguna inversión.

También hay que tener en cuenta que todos los programas y herramientas utilizados en este TFG son gratuitos o en su defecto, se han utilizado con versiones libres. Por ejemplo, la herramienta de visualización Power BI tiene la opción de una versión de pago, sin embargo, se ha utilizado la versión gratuita que aunque con funcionalidad limitada, es más que suficiente para lo que se pretende realizar con ella.

Por tanto, el coste de software es de 0 euros.

Para el coste de personal, se ha utilizado el valor medio de 15 euros que considera la página web https://www.indeed.es, de salario por hora de un Analista-Programador.

Viendo el desglose de tareas realizadas que se encuentra en el último capítulo de esta memoria se puede ver como el total de horas realizadas ha sido de 400 horas.

Es decir, el coste de personal es de 400 x 15=6000 euros

En conclusión, el coste total de este proyecto es la suma de los 3 valores (hardware, software y personal) calculados.

Coste Total= Coste Hardware + Coste Software + Coste de Personal =
76 + 0+ 6000= 6076 euros

El coste ordinario hubiese sido inferior ya que se han realizado lo que se puede considerar horas extras. El coste ordinario se calcula con la previsión inicial de 300 horas para este TFG.

Coste Total Ordinario = Coste Hardware + Coste Software + Coste de Personal Ordinario = 76 + 0 + (15 x 300) = 4576 euros

Hay un sobre coste de 1500 euros que se corresponde con las 100 horas extras.

3 Análisis y especificación de requisitos

3.1 Requisitos funcionales

Definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar ante una determinada entrada y cómo se tiene que comportar ante determinadas situaciones.

- RF-01 El sistema permitirá al usuario seleccionar los periódicos con los que quiere realizar el experimento.
- RF-02 El sistema permitirá al usuario seleccionar el tipo de información con la que realizar el experimento, que extraerá de los periódicos.
- RF-03 El sistema permitirá al usuario introducir el número de titulares que serán guardados en el experimento.
- RF-04 El sistema permitirá al usuario especificar el tiempo que quiere que la herramienta realice el experimento.
- RF-05 El sistema permitirá al usuario salir del programa.
- RF-06 El sistema permitirá al usuario empezar el proceso de extracción de los titulares.
- RF-07 El sistema permitirá al usuario limpiar las selecciones del usuario con las que realizar el experimento.
- RF-08 El sistema permitirá al usuario visualizar los resultados en directo de los titulares almacenados en la base de datos hasta ese momento.
- RF-09 El sistema permitirá al usuario ver un resumen, al final del proceso, de los titulares almacenados en la base de datos.
- RF-10 El sistema permitirá al usuario introducir una serie de palabras para ser buscadas.
- RF-11 El sistema permitirá al usuario introducir una serie de palabras consideradas negativas.
- RF-12 El sistema permitirá al usuario introducir una serie de palabras consideradas positivas
- RF-13 El sistema permitirá al usuario realizar un procesamiento de las palabras introducidas.
- RF-14 El sistema permitirá al usuario la visualización de los titulares que contienen las palabras que previamente a introducido para buscar.
- RF-15 El sistema permitirá al usuario la visualización de los titulares que contienen las palabras negativas que previamente se han introducido.
- RF-16 El sistema permitirá al usuario interrumpir el proceso de extracción de titulares en cualquier momento.
- RF-17 El sistema permitirá al usuario ver una visualización final de los gráficos, al final del proceso, de los titulares almacenados en la base de datos.

- RF-18 Creación de una base de datos con las tablas adecuadas para su posterior uso en otras aplicaciones de visualización.

3.2 Requisitos no funcionales

Un requisito no funcional, especifica cómo debe funcionar el sistema. Pueden llegar a ser más importantes que los propios requisitos funcionales.

Tratan sobre limitaciones que afectan a los servicio o funcionalidades del sistema.

- RNF-01 El sistema permitirá interactuar con el programa de forma fiable, sencilla y rápida.
- RNF-02 El sistema permitirá al usuario ejecutar el programa en distintos terminales.
- RNF-03 El sistema realizará la importación de las palabras negativas se realizará mediante un archivo CSV.
- RNF-04 El sistema realizará la importación de las palabras positivas se realizará mediante un archivo CSV.
- RNF-05 El sistema realizará la importación desde un archivo en formato CSV de las palabras con las que realizar la búsqueda.
- RNF-06 La interfaz de usuario del sistema deberá de ser simple, no requiriendo un gran tiempo de aprendizaje.

3.3 Requisitos de información

- RI-01 El sistema deberá almacenar la información relativa a los nombres de periódicos disponibles para realizar la extracción de titulares.
- RI-02 El sistema deberá almacenar la información relativa a los tipos de información disponibles para la extracción de los titulares.
- RI-03 El sistema deberá almacenar la URL a utilizar para cada periódico y tipo de información disponible.
- RI-04 El sistema deberá almacenar las palabras consideradas negativas por el usuario.
- RI-05 El sistema deberá almacenar las palabras consideradas positivas por el usuario.
- RI-06 El sistema deberá almacenar las palabras con las que realizar una búsqueda proporcionadas por el usuario.

3.4 Casos de uso

3.4.1 Actores

Se considera un único actor, siendo cualquier persona que ejecute y utilice este programa.

3.4.2 Diagrama de casos de uso

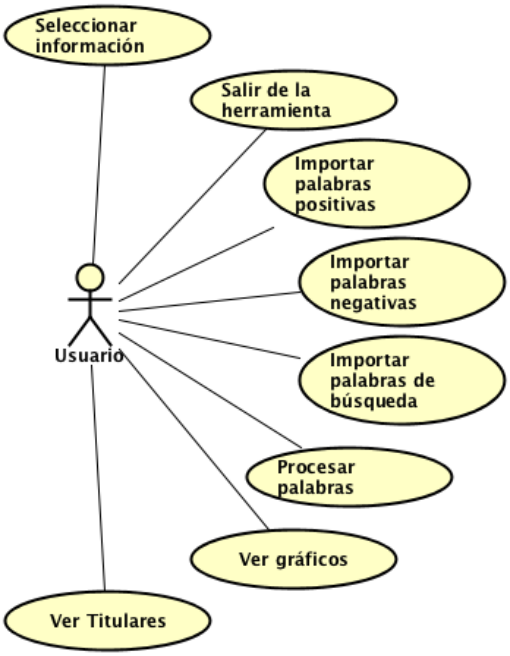


Figura 5. Diagrama casos de uso

3.4.3 Descripción de casos de usos

Caso de uso 1	Seleccionar Información	
Descripción	El sistema deberá permitir al usuario seleccionar la información con la que realizar el proceso de extracción de titulares.	
Precondición	-	
Secuencia Normal	**Paso**	**Acción**
	1	El usuario selecciona los periódicos, tipo de información, método de parada y valor correspondiente.
	2	El sistema verifica que los datos introducidos son correctos y comienza el proceso de extracción.
Excepciones	**Paso**	**Acción**
	2	Si existen campos obligatorios sin introducir o no está en el formato deseado, el sistema informa al usuario y el caso de uso continúa en el paso 1.
Postcondición	• El sistema actualiza la base de datos conforme a las selecciones del usuario.	

Caso de uso 2	Salir de la herramienta	
Descripción	El sistema deberá permitir al usuario salir de la herramienta antes, durante o después del proceso de extracción de titulares.	
Precondición	-	
Secuencia Normal	**Paso**	**Acción**
	1	En cualquier momento el usuario desea abandonar la ejecución de la herramienta pulsando el botón que corresponde.
	2	El sistema permitirá al usuario abandonar y cerrar la herramienta.
Excepciones	**Paso**	**Acción**

	1	Si el usuario desea abandonar la herramienta durante el proceso de extracción de titulares, el sistema mostrará un mensaje de confirmación al usuario de si realmente quiere abandonar el programa. Si el usuario acepta, el caso de uso sigue en el paso 2. Si no lo acepta, continúa en el paso 1.
Postcondición	• El usuario finaliza la ejecución de programa, terminando todos los procesos activos.	

Caso de uso 3	Importar palabras positivas	
Descripción	El sistema deberá permitir al usuario importar un archivo CSV que contenga una lista con las palabras positivas.	
Precondición	• El usuario debe tener un archivo CSV, con el nombre apropiado en la ruta indicada en el manual de usuario. • El proceso de extracción de titulares ha finalizado.	
Secuencia Normal	**Paso**	**Acción**
	1	El actor selecciona la opción de "Importar palabras positivas".
	2	El Sistema lee el archivo CSV proporcionado e informa al usuario de que la tabla correspondiente de la base de datos ha sido actualizada.
Postcondición	• El Sistema actualiza la tabla correspondiente de la base de datos con la lista de palabras positivas facilitada por el usuario.	

Caso de uso 4	Importar palabras negativas	
Descripción	El sistema deberá permitir al usuario importar un archivo CSV que contenga una lista con las palabras negativas.	
Precondición	• El usuario debe tener un archivo CSV, con el nombre apropiado en la ruta indicada en el manual de usuario. • El proceso de extracción de titulares ha finalizado.	
Secuencia Normal	**Paso**	**Acción**
	1	El actor selecciona la opción de "Importar palabras negativas".
	2	El Sistema lee el archivo CSV proporcionado e informa al usuario de que la tabla correspondiente de la base de datos ha sido actualizada.
Postcondición	• El Sistema actualiza la tabla correspondiente de la base de datos con la lista de palabras negativas facilitada por el usuario.	

Caso de uso 5	Importar palabras de búsqueda	
Descripción	El sistema deberá permitir al usuario importar un archivo CSV que contenga una lista con las palabras de búsqueda.	
Precondición	• El usuario debe tener un archivo CSV, con el nombre apropiado en la ruta indicada en el manual de usuario. • El proceso de extracción de titulares ha finalizado.	
Secuencia Normal	**Paso**	**Acción**
	1	El actor selecciona la opción de "Importar palabras de búsqueda".
	2	El Sistema lee el archivo CSV proporcionado e informa al usuario de que la tabla correspondiente de la base de datos ha sido actualizada.

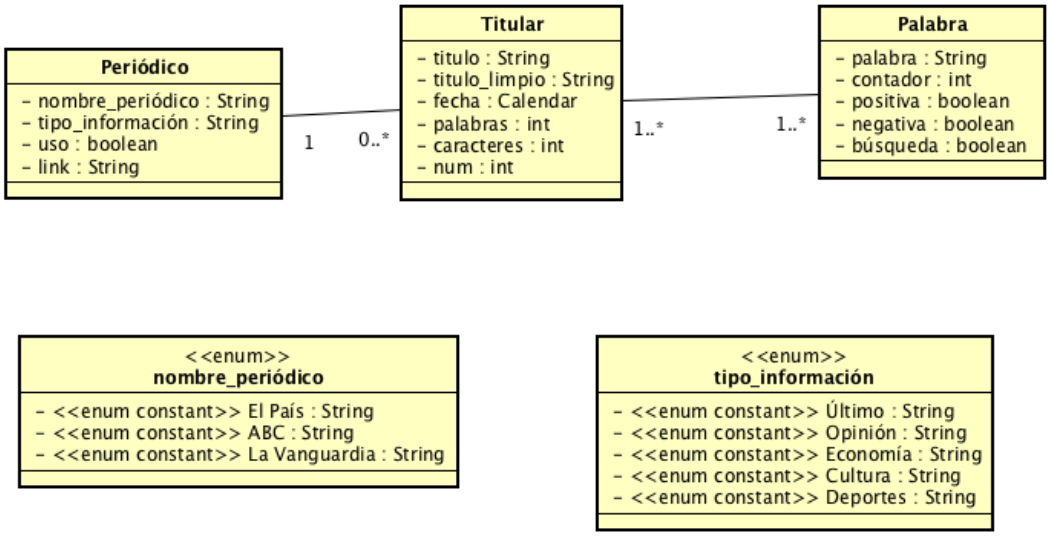
Postcondición	• El Sistema actualiza la tabla correspondiente de la base de datos con la lista de palabras de búsqueda facilitada por el usuario.

Caso de uso 6	Ver gráficos	
Descripción	El sistema deberá permitir al usuario la visualización de los gráficos con las estadísticas de la información almacenada en la base de datos hasta ese momento.	
Precondición	• El proceso de extracción de titulares ha finalizado, es decir, la base de datos se encuentra actualizada.	
Secuencia Normal	**Paso**	**Acción**
	1	El usuario selecciona la opción de "Ver Gráficos".
	2	El sistema muestra los gráficos con la información disponible en ese momento en la base de datos.
Postcondición	• El sistema mostrará los gráficos con las estadísticas de los titulares almacenados en la base de datos.	

Caso de uso 7	Ver titulares	
Descripción	El sistema deberá permitir al usuario ver los titulares que contengan una palabra de búsqueda o hayan sido considerados negativos o positivos.	
Precondición	• El proceso de extracción de titulares ha finalizado. • El procesamiento de palabras se ha realizado.	
Secuencia Normal	**Paso**	**Acción**
	1	El usuario selecciona la opción de "Ver Titulares".
	2	El sistema muestra los titulares correspondientes, almacenados en ese momento en la base de datos.
Postcondición	El sistema mostrará una tabla con los titulares que contienen una palabra de búsqueda, o han sido considerados positivos y negativos	

Caso de uso 8	Procesar palabras	
Descripción	El sistema deberá permitir al usuario realizar un procesamiento de palabras de búsqueda, positivas y negativas.	
Precondición	El proceso de extracción de titulares ha finalizado. Se han importado los archivos CSV correspondientes, actualizando la base de datos.	
Secuencia Normal	**Paso**	**Acción**
	1	El usuario selecciona la opción de "Procesamiento de palabras".
	2	El Sistema procesa las palabras de la base de datos e informa al usuario de que la base de datos ha sido actualizada.
Postcondición	• La base de datos se actualiza con la información proporcionada por el usuario.	

3.5 Modelo de dominio



En este proyecto, existen tres tipos de clases:

- Periódico
    - nombre_periódico: Periódico del que se obtienen los titulares. Valores especificados en el <<enum>>.
    - tipo_información: Clase de noticia, dónde se incluye el titular. Valores especificados en el <<enum>>.
    - uso: Valor para especificar si el usuario lo ha seleccionado o no para la extracción de titulares.
    - link: URL del RSS
- Titular
    - titulo: titular publicado obtenido del RSS
    - titulo_limpio: titular al que se le han eliminado las palabras con menor importancia
    - fecha: Fecha de publicación del titular
    - palabras: Número de palabras que forman el titular
    - caracteres: Número de caracteres que forman el titular
    - num: Identificador numérico de cada titular.
- Palabra
    - palabra: Cada una de las palabras, que forman el titulo_limpio.
    - contador: Veces que aparece la palabra en un titular.
    - positiva: indicador de si la palabra es considerada positiva
    - negativa: indicador de si la palabra es considera negativa
    - búsqueda: indicador de se la palabra es de búsqueda

- Cardinalidad:

    - Un titular pertenecerá a un único periódico.
    - Un periódico puede tener varios titulares o ninguno.
    - Un titular puede tener una o varias palabras.
    - Una palabra puede estar en uno o varios titulares.

3.6 Diagramas de secuencia

Diagramas de secuencia de los casos de uso más importantes de la herramienta.

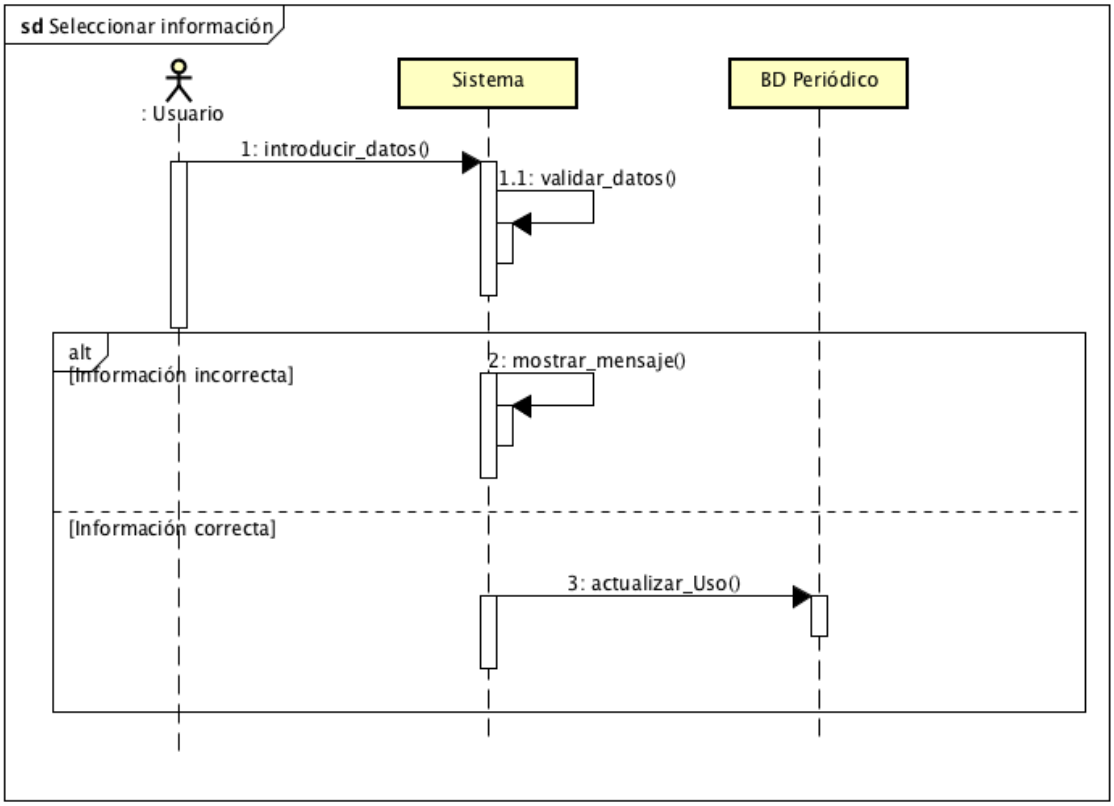


Figura 6. Diagrama de secuencia Seleccionar información

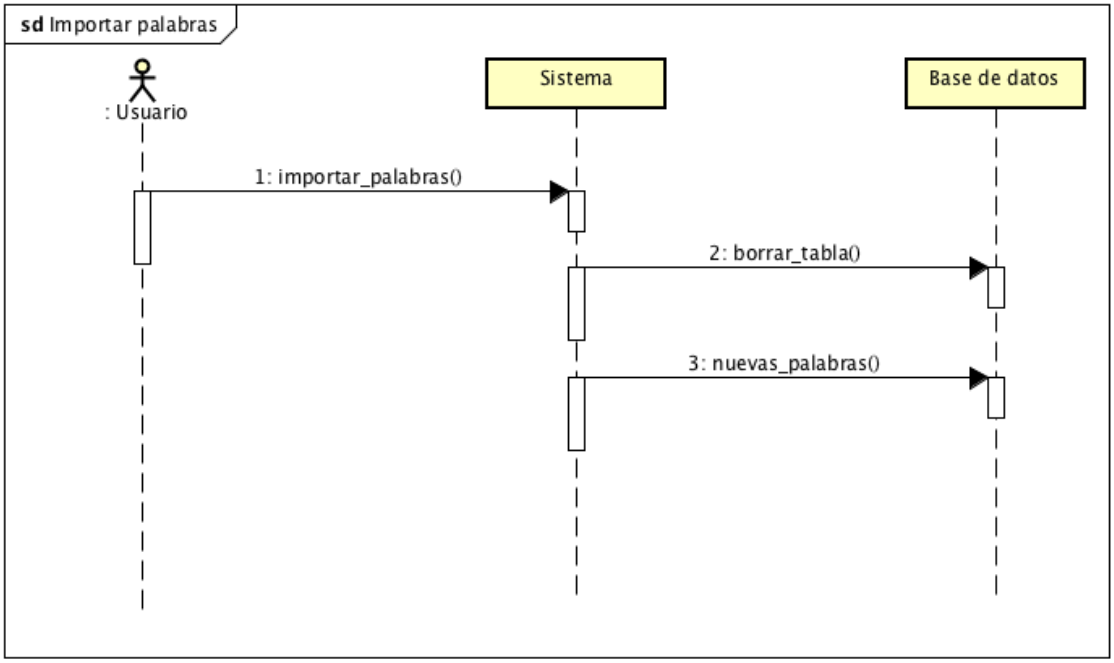


Figura 7. Diagrama de secuencia Importar palabras

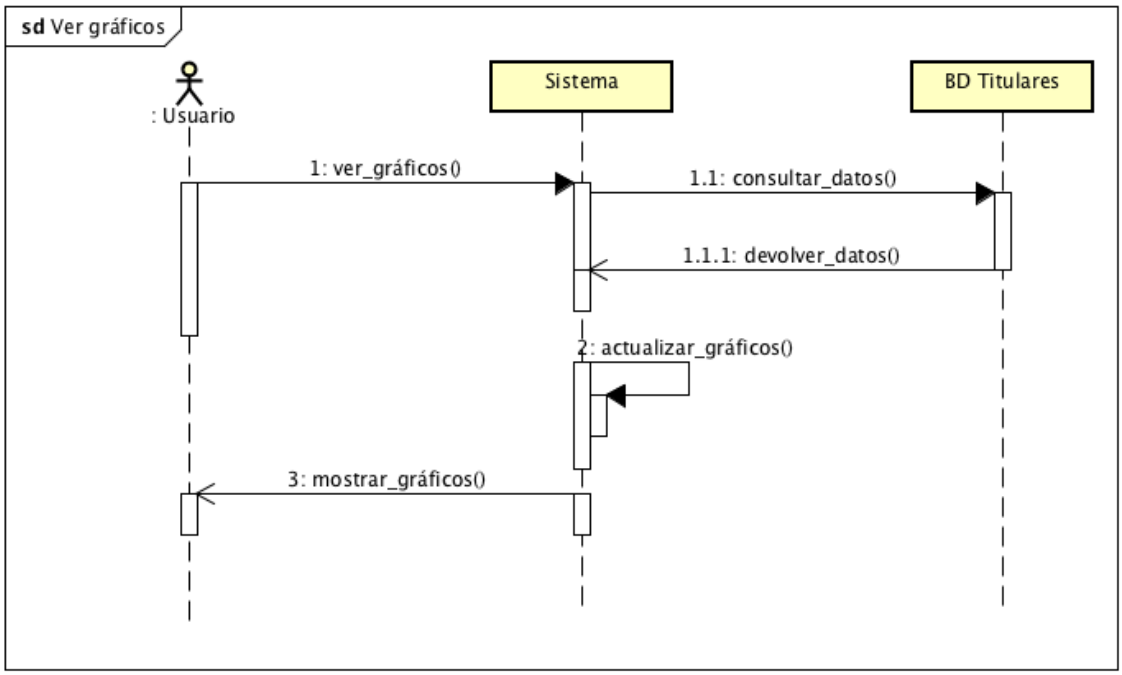


Figura 8. Diagrama de secuencia Ver gráficos

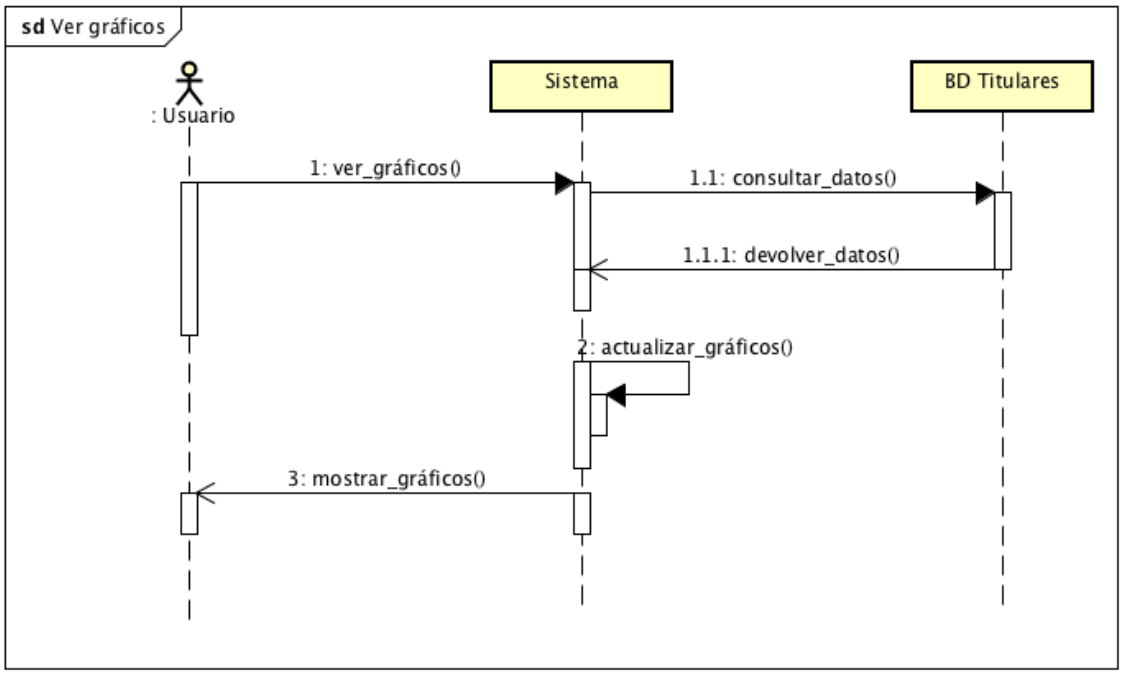


Figura 9. Diagrama de secuencia Ver titulares

4 Herramientas utilizadas

4.1 Lenguaje Java

Java (Wikipedia, 2019) es un lenguaje de programación de propósito general, concurrente, orientado a objetos. Se diseñó con el fin de tener las mínimas dependencias de implementación posibles.

Tiene la intención de permitir a los desarrolladores de aplicaciones que escriban el programa, y pueda ser ejecutado en cualquier dispositivo. Esto se conoce en inglés como WORA: "Write once, run anywhere". Significa que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra.

Java es uno de los lenguajes de programación más populares en uso, especialmente para aplicaciones cliente-servidor de web.

Java se creó con cinco objetivos principales:

1) Debería usar el paradigma de la programación orientada a objetos.
2) Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3) Debería incluir por defecto soporte para trabajo en red.
4) Debería diseñarse para ejecutar código en sistemas remotos de forma segura,
5) Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos como C++.

Razones de utilización del lenguaje de programación Java:

- Lenguaje de programación con el que el alumno tiene mayores conocimientos y experiencia.
- Código base encontrado en este lenguaje de programación.
- Capacidad para desarrollar una interfaz de usuario.

Figura 10. Logo Java

4.2 NetBeans

NetBeans (Wikipedia, 2019) es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software denominados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial "manifest file", que lo identifica como módulo.

Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans, pueden ser extendidas fácilmente por otros desarrolladores de software.

Características de la aplicación:

- Gestión de la interfaz de usuario (menús y barras de herramientas).
- Gestión de configuración de usuario.
- Gestión de almacenamiento (guardar o cargar algún tipo de dato).
- Gestión de ventana.
- Marco Asistente.
- Librería visual.
- Herramientas de desarrollo integrado.

Razones de utilización de NetBeans:

- Necesidad de utilización de un IDE, ya que proporciona muchas ventajas en cuanto a rapidez y facilidad para la programación.
- IDE con el que el alumno tiene mayores conocimientos y experiencia.
- No es necesario invertir tiempo en su aprendizaje.
- Posee la capacidad de desarrollar una interfaz de usuario paso a paso.

Figura 11. Logo NetBeans

4.3 Java Swing

Java Swing (Wikipedia, 2019) es un framework MVC para desarrollar interfaces gráficas para Java con independencia de la plataforma. Sigue un simple modelo de programación por hilos, y posee las siguientes características principales:

- Independencia de plataforma.
- Extensibilidad: Arquitectura altamente particionada: los usuarios pueden proveer sus propias implementaciones modificadas para sobrescribir las implementaciones por defecto.
- El control permite representar diferentes estilos de apariencia "look and feel" (desde apariencia Mac OS hasta apariencia Windows XP). Además, los usuarios pueden proveer su propia implementación de apariencia, que permitirá cambios uniformes en la apariencia existente en las aplicaciones Swing sin efectuar ningún cambio al código de aplicación.

Razones de utilización de Java Swing:

- Necesidad de diseñar una interfaz gráfica.
- Con la utilización del IDE NetBeans, facilitaba la creación de la UI.

4.4 MySQL

MySQL es un sistema de gestión de base de datos relacional desarrollado por Oracle Corporation. Considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos Web. Está desarrollado en su mayor parte en ANSI C y C++.

Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta, es decir, cualquier persona puede bajar el código fuente y usarlo si pagar, así como ajustarlo a sus necesidades.

Características:

- Tablas hash en memorias temporales.
- Completo soporte para operadores y funciones en cláusulas SELECT y WHERE.
- Completo soporte para cláusulas GROUP BY y ORDER BY, soporte de funciones agrupación.
- Ofrece un sistema de contraseñas y privilegios.
- Soporta gran cantidad de datos. Bases de datos de hasta 50 millones de registros.
- Permite hasta 64 índices por tabla.
- Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas.
- Máximo ancho de límite son 1000 bytes.
- Clientes se conectan con el servidor MySQL usando sockets TCP/IP en cualquier plataforma.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Transacciones y claves foráneas.
- Búsqueda e indexación de campos de texto.

Razones de utilización de MySQL:

- Sistema de gestión de bases de datos con el que el alumno tiene mayores conocimientos y experiencia.
- No es necesario invertir tiempo en su aprendizaje.

(Dri, 2019)

Figura 12. Logo MySQL

4.5 MySQL WorkBench

MySQL WorkBench (Wikipedia, 2019) es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.

Características:

- Editor de SQL
  - Exploración de esquema de objetos.
  - Resaltado de sintaxis en SQL y analizador de declaraciones.
  - Conjunto de resultados múltiples, editables.
  - Tunenilazión de conexión por SSH.
- Modelado de datos:
  - Diagrama entidad-relación.
  - Modelado visual con arrastrar y soltar.
- Administración de base de datos:
  - Iniciar y detener instancias de bases de datos.
  - Administración de cuentas en base de datos.
  - Exploración de instancias variables.
  - Exploración de ficheros de registros.
  - Exportación e importación masiva de datos.

Razones de utilización de MySQL WorkBench:

- Debido a la gran cantidad de información almacenada en la base de datos, necesidad de manejo de una base de datos más allá del terminal del ordenador.
- Facilita las operaciones de consulta, inserción y borrado de observaciones de las distintas tablas.

(Macupdate, 2019)

Figura 13. Logo MySQL WorkBench

4.6 Astah UML

Astah UML (Scribd, 2019) es una herramienta de diseño de sistemas que soporta UML, Diagrama de Relación de Entidades, diagramas de flujo, CRUD…

Utilidades de Astah UML:

- Ver proyecto.
- Mostrar la estructura de los modelos.
- Mostrar la estructura de la herencia de clases.
- Editor de diagramas.
- Mostrar lista de diagramas en el proyecto.

Razones de utilización de Astah UML:

- Programa más utilizado a lo largo de la carrera para la representación modelo de dominio, casos de uso, diagramas de secuencia…

(Astah, 2019)

Figura 14. Logo Astah UML

4.7 Git

Git (Wikipedia, 2019) es un software de control de versiones, diseñado pensando en la eficiencia y la contabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Tiene como propósito llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Características:

- Apoyo al desarrollo no lineal. Rapidez en la gestión de ramas y mezclado de versiones. Incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal.
- Gestión distribuida. Git proporciona a cada programado una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales
- Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
- Todas las versiones previas a un cambio de terminado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio.
- Realmacenamiento periódico en paquetes (ficheros).

Razones de utilización de git:

- Necesidad de llevar un control de versiones.
- Permite la utilización del programa en varios ordenadores, con una gran facilidad.
- Mejor método para el control de versiones existente en la actualidad.

(Wikipedia, 2019)

Figura 15. Logo git

4.8 Microsoft Project

Microsoft Project (MSP) (WIkipedia, 2019) es un software de administración de proyectos y programas de proyectos, diseñado, desarrollado y comercializado por Microsoft. Se ideó para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

Razones de utilización de Microsoft Project:

- Herramienta utilizada durante la carrera para la creación de diagramas de Gantt, para visualizar la planificación de un proyecto.

(Wikipedia, 2019)

Figura 16. Logo Microsoft Project

4.9 Lenguaje R

R (Wikipedia, 2019) es un entorno y lenguaje de programación con un enfoque al análisis estadístico. Se trata de uno de los lenguajes de programación más utilizados en investigación científica, siendo además muy popular en el campo de la minería de datos, investigación biomédica, la bioinformática y las matemáticas financieras. Se debe a la posibilidad de cargar diferentes bibliotecas o paquetes con funcionalidades de cálculo y creación de gráficas . R proporciona un amplio abanico de herramientas estadísticas y gráficas

Además, R puede integrarse con distintas bases de datos, así como la existencia de bibliotecas que facilitan su utilización desde lenguajes de programación interpretados.

Otras de las características de R es su capacidad gráfica que permite generar gráficos con alta calidad. Posee su propio formato para la documentación basado en LaTeX.

Razones de utilización de Lenguaje R:

- Lenguaje estadístico con gran capacidad para la creación de visualizaciones gráficas.
- Permite enseñar una de las capacidades de la herramienta conectándolo con un lenguaje estadístico.

(Maximaformacion, 2019)

Figura 17. Logo lenguaje R

4.10 RStudio

RStudio (Wikipedia, 2019) es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo.

Características:

- Resaltado de sintaxis, auto-completado de código y sangría inteligente.
- Ejecutar código R directamente desde el editor de código fuente.
- Salto rápido a las funciones definidas.
- Documentación y soporte integrado.
- Administración sencilla de múltiples directorios de trabajo mediante proyectos.

- Navegación en espacios de trabajo y visor de datos.
- Depurador interactivo para diagnosticar y corregir los errores rápidamente.

Razones de utilización de RStudio:

- IDE más conocido y utilizado para el lenguaje R.
- Como cualquier otro IDE, tiene muchas ventajas que facilitan al desarrollador la programación de la herramienta.
- Posibilidad de crear un reporte que contenga los gráficos creados, como ejemplo de uso de la herramienta en su conjunto.

(Proyectosbeta, 2016)

Figura 18. Logo RStudio

4.11 Power Bi Desktop

Power Bi Desktop (Microsoft, 2019) en una solución de análisis empresarial que permite visualizar los datos de forma interactiva.

Permite crear una colección de consultas, conexiones de datos e informes que se pueden compartir fácilmente con otros usuarios. Integra tecnologías de eficacia comprobada de Microsoft (un potente motor de consultas, capacidades de modelado de datos y visualizaciones).

Se trata de una herramienta eficaz, flexible y muy accesible para conectarse con datos y darles forma, crear modelos eficaces y elaborar informes con la estructura adecuada.

Power Bi Desktop centraliza, simplifica y agiliza lo que de otro modo podría ser un proceso de diseño y creación de repositorios e informes de inteligencia empresarial disperso, arduo y desconectado.

Razones de utilización de Power Bi Desktop:

- Necesidad de mostrar resultados con un programa más potente en cuanto a la capacidad de visualización de gráficos se refiere.
- Gran disponibilidad de tipos de gráficos.
- Posibilidad de creación de gráficos y dashboards interactivos.

(Isaacfigueroa, 2019)

Figura 19. Logo Power BI

5 Diseño

En este capítulo se llevará a cabo la explicación de diseño utilizado en el desarrollo de la aplicación para el cumplimiento de los requisitos mencionados anteriormente.

5.1 Arquitectura

La arquitectura de software (Wikipedia, 2019) consiste en el diseño de más alto nivel de la estructura en un sistema.

- Consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco claro y definido de interacción con el código fuente del software.

- La arquitectura de software es diseñada y seleccionada en función de los requisitos y restricciones del sistema.

- En función de la tecnología empleada se optará por uno u otro tipo de arquitectura.

- Con la arquitectura de software se define, de manera abstracta, los componentes que se llevan a cabo con las tareas de computación, interfaces, así como la comunicación entre ellos.

Para la realización de este TFG se ha utilizado el patrón Modelo-Vista-Controlador (MVC).

El MVC (Wikiepdia, 2019) es un patrón de arquitectura de software, que consiste en separar los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y comunicaciones.

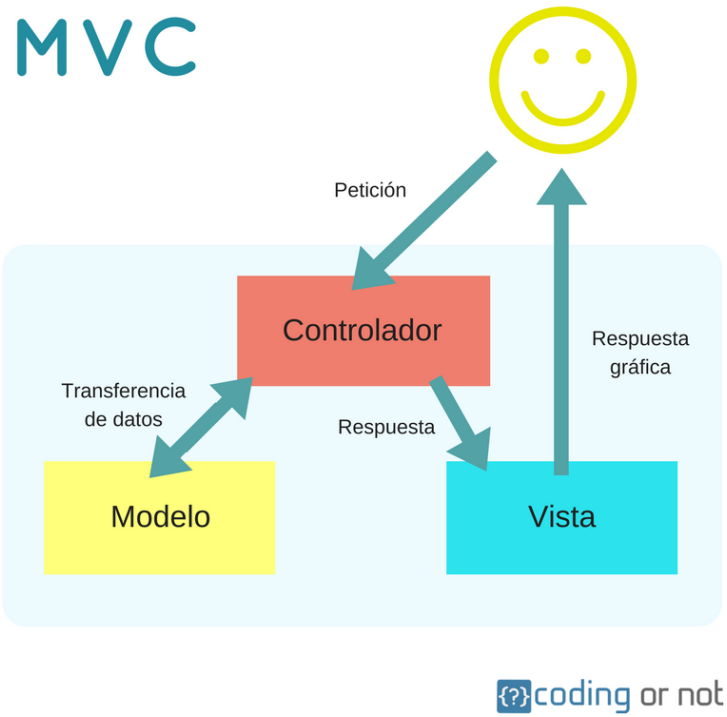
Consta de tres componentes distintos que son el modelo, la vista y el controlador. Define por una parte los componentes para la representación de la información, y por otra la interacción del usuario.

Explicación de cada uno de los componentes:

- **Modelo**. Se trata de la representación de la información con la que el sistema funciona. Conlleva los accesos a dicha información, tanto consultas como actualización de la misma. Implementa los privilegios de acceso a la misma.
  Envía a la "vista", la información solicitada para ser mostrada al usuario.
  Recibe peticiones de acceso a la información por parte del "controlador".

- **Vista**. Es la presentación que se realiza al usuario del "modelo", mediante una interfaz de usuario.
  Necesita que el "modelo" proporcione la información solicitada.

- **Controlador**. Encargado de responder a cada una de las acciones que realiza el usuario. Realiza una petición de información al "modelo". Además, envía comandos a la "vista" para un cambio en la representación de la misma.
  Es el intermediario entre la "vista" y el "modelo".



(Codingornot, 2019)

Figura 20. Modelo-Vista-Controlador

Flujo de control que se sigue con el MVC:

1) El usuario interacciona con la interfaz.

2) El controlador recibe la notificación de la acción solicitada por el usuario. Se gestiona el evento.

3) El controlador accede al modelo, realizando las modificaciones necesarias en función de la petición del usuario.

4) El controlador delega en la vista la representación de la interfaz de usuario, con los cambios en el modelo y la información obtenida del mismo.

5) La interfaz de usuario espera nuevas interacciones del usuario para comenzar el ciclo de nuevo.

En este TFG, se utiliza un Sistema de gestión de Base de Datos, el cuál gestiona los datos que debe utilizar la aplicación. Dentro del MVC, esta gestión se encuentra dentro del modelo.

5.2 Uso de MVC

El desarrollo de este TFG consta de los siguientes 5 paquetes:

1) Vista.
2) Controlador.
3) Modelo.
4) Principal.
5) Levenshtein.

A continuación, se procede a explicar el contenido de cada uno de los paquetes:

5.2.1 Vista

Está formado por 5 clases java, uno por cada una de las 5 vistas que forman en proyecto.

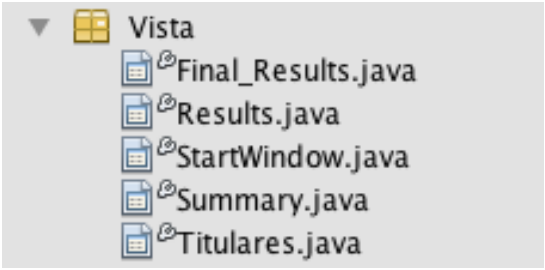


Figura 21. Paquete Vista

Siguiendo el patrón MVC, en estas clases sólo se encuentra el contenido de los distintos componentes que forman la interfaz. No hay ninguna función ni procedimiento más allá de lo puramente visual.

5.2.2 Controlador

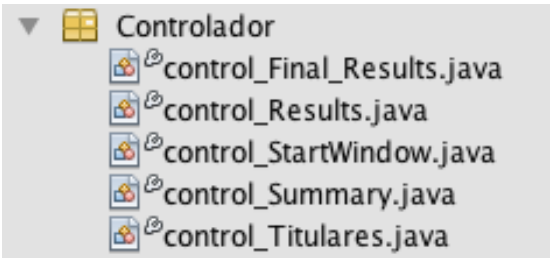


Figura 22. Paquete Controlador

Se encuentran las 5 clases javas que se corresponden con los 5 controladores de cada una de las vistas.

La siguiente imagen, es un ejemplo de una de ellas:

```
public class control_Summary implements ActionListener{
 private UsoBD mod;
 public Summary miVista;
 public Periodico period;

 public control_Summary(Periodico period,UsoBD mod ,Summary miVista){
        this.mod=mod;
        this.period=period;
        this.miVista=miVista;
        this.miVista.Graficos.addActionListener(this);
        this.miVista.Procesamiento.addActionListener(this);
        this.miVista.Importar_Negativas.addActionListener(this);
        this.miVista.Importar_Busqueda.addActionListener(this);
        this.miVista.Titulares.addActionListener(this);

    }
```

Figura 23. Ejemplo Controlador

Se puede apreciar como la clase implementa la clase ActionListener para poder determinar cuando el usuario realiza una acción con un elemento de la vista correspondiente.

Cuando el usuario interacciona con la vista, el controlador captura ese evento y en función de cuál sea realiza la acción correspondiente, como se puede ver a continuación:

```
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == miVista.Graficos) {
        finalizar();
    }
    if (e.getSource() == miVista.Procesamiento) {
        procesar();
        JOptionPane.showMessageDialog(null,"Procesamiento finalizado");
    }
```

Figura 24. Ejemplo captura de evento

Se puede apreciar como dependiendo del evento que se capture se llama por ejemplo a un método o se muestra un texto en pantalla.

5.2.3 Modelo

Figura 25. Paquete Modelo

5.2.3.1 Conexión

Clase encargada de realizar la conexión de NetBeans con la base de datos local de MySQL. En esta clase es especifican todos lo datos necesarios para realizar la conexión como son el usuario, el nombre y la contraseña de la base de datos.

Posee un método para realizar la conexión y otro para desconectarse. De esta forma se evita dejar la conexión abierta con la base de datos.

5.2.3.2 FeedMessage

Clase que se encarga de almacenar la información de cada una de las instancias que se corresponden con los titulares. Sus atributos son los que posteriormente se guardarán en cada una de las columnas de la base de datos de la tabla Titulares.

Se incluyen métodos getters() y setters() para la obtención y modificación de estos atributos según sea necesario.

```
public class FeedMessage {

    String title;
    String titulo_limpio;
    String campo_fecha;
    Calendar fecha;
    int palabras;
    int caracteres;
    String nombre_periodico;
    String tipo_informacion;
    int num;
```

Figura 26. Atributos FeedMessage

5.2.3.3 Palabras

Clase que se encarga de almacenar la información de cada una de las instancias que se corresponden con las palabras. Sus atributos son los que posteriormente se guardarán en cada una de las columnas de la base de datos de la tabla Palabras.

Se incluyen métodos getters() y setters() para la obtención y modificación de estos atributos según sea necesario.

```
public class Palabras {

    String palabra;
    String nombre_periodico;
    String tipo_informacion;
    int contador;
    int num;
    boolean negativa;
    boolean busqueda;
```

Figura 27. Atributos Palabras

5.2.3.4 Periódico

```
public class Periodico {

    String nombre_periodico;
    String tipo_informacion;
    String link;
    boolean uso;
```

Figura 28. Atributos periódico

Clase que se encarga de almacenar la información de cada una de las instancias que se corresponden con los periódicos. Sus atributos son los que posteriormente se guardarán en cada una de las columnas de la base de datos de la tabla Periódico.

Se incluyen métodos getters() y setters() para la obtención y modificación de estos atributos según sea necesario.

5.2.3.5 UsoBD

Clase que donde se realiza toda la funcionalidad relacionada con la base de datos. Esta funcionalidad incluye la realización de las consultas necesarias, la inserción de filas a la tabla de Titulares y la inserción de filas a la tabla de Palabras.

5.2.4 Principal

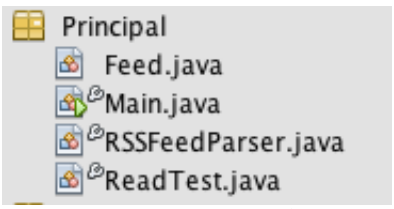


Figura 29. Paquete Principal

Lo forman las clases necesarias para realizar el proceso de extracción de titulares, es decir, para conectarse a la URL proporcionada y leer el archivo XML.

5.2.5 Levenshtein

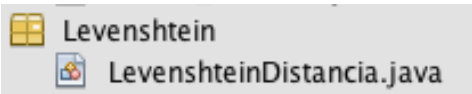


Figura 30. Paquete Levenshtein

Clase para calcular la similitud de dos cadenas de caracteres, mediante la distancia de Levenshtein.

5.3 Diseño base de datos relacional

5.3.1 Diagrama relacional

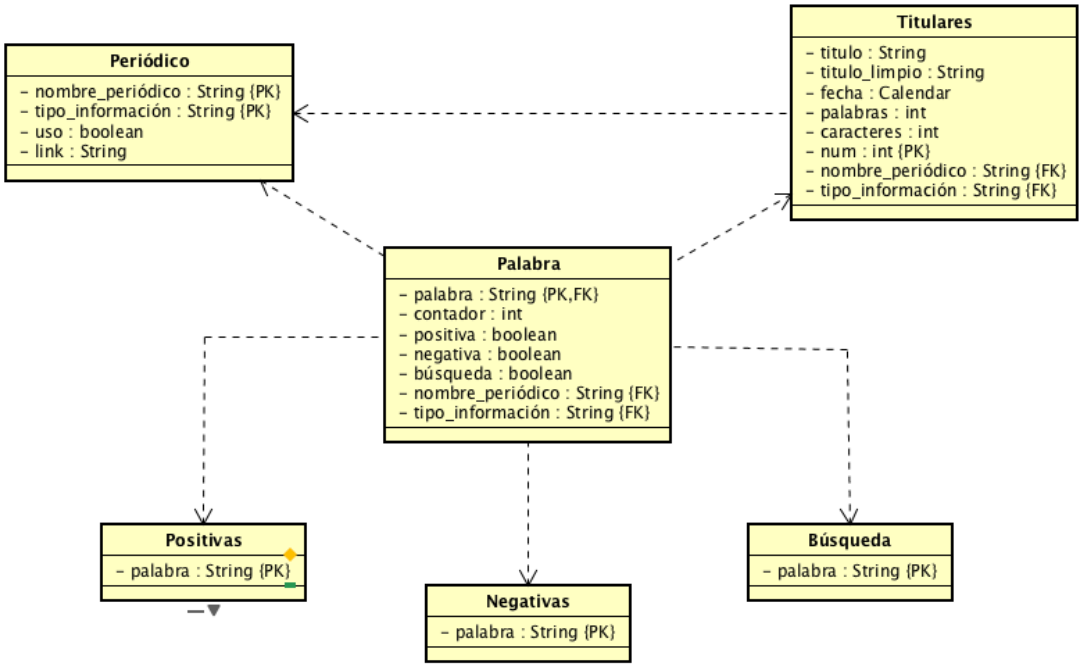


Figura 31. Diagrama relacional de la base de datos.

5.3.2 Interacciones con la base de datos

A lo largo de la ejecución de la herramienta se realizan una serie de interacciones con la base de datos. Estas interacciones son consultas a tablas determinadas de la base de datos, inserción de filas y actualización de filas.

A continuación, se muestran las interacciones más importantes que se da a la base de datos, desde la herramienta.

1) Almacenar titulares:

```
INSERT INTO Titulares (titulo,titulo_limpio,fecha,palabras,
caracteres,nombre_periodico,tipo_informacion, num)
VALUES (?, ?, ?, ?, ?, ?, ?, ?);
```

Figura 32. SQL Almacenar titulares

Para insertar cada uno de los titulares con su información correspondiente en la tabla de Titulares.

2) Guardados desde ahora:

```
SELECT COUNT(*) as total
FROM Titulares
where fecha > "hora_inicio";
```

Figura 33. SQL Guardados desde ahora

Determina el número de titulares con fecha de publicación posterior a la fecha de inicio de la ejecución del programa. Se encarga de cumplir la condición de parada del programa con el número de titulares especificado por el usuario.

3) Numero de titulares en cada periódico:

```
SELECT COUNT(*) as total
FROM Titulares
WHERE nombre_periodico=?;
```

Figura 34. SQL Número de titulares en cada periódico

Devuelve el número de titulares almacenados del periódico proporcionado. Sirve para poder conocer el número de periódicos almacenados en el resumen de resultados, tras la finalización del proceso de extracción de titulares.

4) Último titular almacenado:

```
SELECT *
FROM Titulares
WHERE num = getUlt();
```

Figura 35. SQL Último titular almacenado

Devuelve el último titular que ha sido almacenado proporcionando su identificador. El identificador del último titular almacenado es el número de titulares almacenados. Este titular es el que se mostrará en la vista de resultados en directo, actualizándose a medida que se almacena un nuevo titular.

5) Búsqueda de una palabra específica:

```
SELECT *
FROM Palabras
WHERE palabra=? AND nombre_periodico=?
AND tipo_informacion =? AND num =?;
```

Figura 36. SQL Búsqueda de una palabra específica

Búsqueda de una palabra de un titular en concreto para ver si ya está almacenado en la tabla de datos o no. En los casos en que en un mismo titular aparezca repetida una misma palabra, esta búsqueda verá que ya está almacenada y hay que aumentar el contador de esa palabra, para ese titular.

6) Tabla titulares agrupados:

```
SELECT nombre_periodico,tipo_informacion, COUNT(*)
FROM Titulares
WHERE nombre_periodico=?
GROUP BY tipo_informacion;
```

Figura 37. SQL Tabla titulares agrupados

Devuelve una tabla con el número de titulares almacenados para cada uno de los tres periódicos.

7) Tabla media de palabras agrupadas:

```
SELECT nombre_periodico,tipo_informacion, AVG(palabras)
FROM Titulares
WHERE nombre_periodico=?
GROUP BY tipo_informacion;
```

Figura 38. SQL Tabla media de palabras agrupadas

Devuelve una tabla con el promedio de palabras de los titulares almacenados para cada uno de los tres periódicos.

8) Tabla media de caracteres agrupados:

```
SELECT nombre_periodico,tipo_informacion, AVG(caracteres)
FROM Titulares
WHERE nombre_periodico=?
GROUP BY tipo_informacion;
```

Figura 39. SQL Tabla media de caracteres agrupados

Devuelve una tabla con el promedio de caracteres de los titulares almacenados para cada uno de los tres periódicos.

9) Titulares con una palabra de búsqueda:

```
SELECT titulo,fecha,titulares.nombre_periodico, titulares.tipo_informacion
FROM Titulares, Palabras
WHERE palabras.num=titulares.num AND busqueda=TRUE;
```

Figura 40. SQL Titulares con una palabra de búsqueda

Devuelve los titulares que contienen al menos una de las palabras consideradas como de búsqueda.

10) Titulares negativos:

```
SELECT titulo
FROM Titulares ,
(SELECT num, sum(negativa) as nega, sum(positiva) as pos
FROM palabras GROUP BY num) as myalias
WHERE nega > pos  AND myalias.num=Titulares.num;
```

Figura 41. SQL Titulares negativos

Devuelve los titulares que han sido considerados negativos. Contienen más palabras negativas que positivas.

11) Titulares positivos

```
SELECT titulo
FROM Titulares ,
(SELECT num, sum(negativa) as nega, sum(positiva) as pos
FROM palabras GROUP BY num)as myalias
WHERE nega < pos AND myalias.num=Titulares.num;
```

Figura 42. SQL Titulares positivos

Devuelve los titulares que han sido considerados negativos. Contienen más palabras positivas que negativas.

12) Actualizar uso de URL:

```
UPDATE Periodico SET uso=TRUE
WHERE nombre_periodico=? AND tipo_informacion=?
```

Figura 43. SQL Actualizar uso de URL

Cambia el campo de uso a true, de la URL del periódico u tipo de información proporcionados.

13) Procesar palabra negativa:

```
UPDATE Palabras SET negativa=TRUE WHERE palabra=?
```

Figura 44. SQL Procesar palabra negativa

Cambia el campo de negativa a true, de la palabra proporcionada.

14) Resetear uso de URL:

```
UPDATE Periodico SET uso=FALSE
```

Figura 45. SQL Resetear uso de URL

Cambia los campos de uso de todas las URLs a false. Se realza antes de actualizar la nueva selección del usuario.

15) Información de URL para utilizar:

```
SELECT nombre_periodico, tipo_informacion, link
FROM Periodico
WHERE uso=TRUE;
```

Figura 46. SQL Información de URL para utilizar

Devuelve aquellas URLs que tienen el campo de uso a true. Son los links que el usuario ha seleccionado para realizar la extracción de titulares.

16) Almacenar palabras:

```
INSERT INTO Palabras (palabra,nombre_periodico,tipo_informacion,
contador,num,negativa,positiva,busqueda)
 VALUES (?, ?, ?, ?, ?, ?, ?, ?)
```

Figura 47. SQL Almacenar palabras

Para insertar cada uno de las palabras con su información correspondiente en la tabla de Palabras.

17) Frecuencia de cada palabra:

```
SELECT palabra, suma
FROM (SELECT palabra ,sum(contador) as suma
FROM Palabras GROUP BY palabra) as myalias
WHERE suma>1;
```

Figura 48. SQL Frecuencia de cada palabra

Obtiene la frecuencia de cada una de las palabras almacenadas en la base de datos. Solo para las palabras que aparecen más de una vez.

18) Importar CSV de palabras negativas:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/negativas.csv'
INTO TABLE negativas FIELDS TERMINATED BY ',' LINES TERMINATED BY '\\n';
```

Figura 49. SQL Importar CSV de palabras negativas

Importa el archivo CSV de palabras negativas a la tabla correspondiente.

19) Palabras distintas:

```
SELECT palabra, suma
FROM (SELECT palabra ,sum(contador) as suma
FROM Palabras GROUP BY palabra) as myalias;
```

Figura 50. SQL Obtener palabras distintas

Devuelve una lista con todas las palabras distintas. Esto se debe a que en la tabla de palabras se guardan palabras repetidas, ya que se almacenan todas las palabras de cada titular haciendo referencia al titular al que pertenecen. Con esta consulta se obtienen una lista de palabras que se correspondería a una especie de diccionario de palabras almacenadas.

5.3.3 Script SQL

En este apartado se muestra el script utilizado para la creación de las tablas correspondientes de la tabla de datos.

```
CREATE TABLE Periodico(
    nombre_periodico CHAR(255),
    tipo_informacion CHAR(255),
    link CHAR(255),
    uso BOOLEAN,
    PRIMARY KEY (nombre_periodico,tipo_informacion)
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
CREATE TABLE Titulares(
    titulo CHAR(255),
    titulo_limpio CHAR(255),
    fecha DATETIME,
    palabras int,
    caracteres int,
    nombre_periodico CHAR(100),
    tipo_informacion CHAR(100),
    num int,
    PRIMARY KEY (num),
    FOREIGN KEY (nombre_periodico,tipo_informacion)
    REFERENCES Periodico(nombre_periodico,tipo_informacion)
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Figura 51. Script SQL 1

```
CREATE TABLE Palabras(
    palabra CHAR(255),
    nombre_periodico CHAR(255),
    tipo_informacion CHAR(255),
    contador int,
    num int,
    negativa BOOLEAN,
    busqueda BOOLEAN,
    PRIMARY KEY (palabra,nombre_periodico,tipo_informacion,num),
    FOREIGN KEY (nombre_periodico,tipo_informacion)
    REFERENCES Periodico(nombre_periodico,tipo_informacion),
    FOREIGN KEY (num) REFERENCES Titulares (num)
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
CREATE TABLE Negativas(
    palabra CHAR(255),
    PRIMARY KEY (palabra),
    FOREIGN KEY (palabra) REFERENCES Palabras(palabra)
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
CREATE TABLE Busqueda(
    palabra CHAR(255),
    PRIMARY KEY (palabra),
    FOREIGN KEY (palabra) REFERENCES Palabras(palabra)
)ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

Figura 52. Script SQL 2

```
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'El Pais', 'Último','http://ep00.epimg.net/rss/tags/ultimas_noticias.xml',false );
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'El Pais', 'Opinión','http://ep00.epimg.net/rss/elpais/opinion.xml' ,false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'El Pais', 'Economía','http://ep00.epimg.net/rss/economia/portada.xml',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'El Pais', 'Cultura', 'http://ep00.epimg.net/rss/cultura/portada.xml',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'El Pais', 'Deportes', 'http://ep00.epimg.net/rss/deportes/portada.xml',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'ABC', 'Último', 'https://www.abc.es/rss/feeds/abc_ultima.xml',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'ABC', 'Opinión','https://www.abc.es/rss/feeds/abc_opinioncompleto.xml' ,false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'ABC', 'Economía', 'https://www.abc.es/rss/feeds/abc_Economia.xml',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
```

Figura 53. Script SQL 3

```
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'ABC', 'Cultura','https://www.abc.es/rss/feeds/abc_Cultura.xml' ,false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'ABC', 'Deportes', 'https://www.abc.es/rss/feeds/abc_Deportes.xml',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'La Vanguardia', 'Último', 'https://www.lavanguardia.com/mvc/feed/rss/home',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'La Vanguardia', 'Opinión', 'https://www.lavanguardia.com/mvc/feed/rss/opinion',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'La Vanguardia', 'Economía', 'https://www.lavanguardia.com/mvc/feed/rss/economia',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'La Vanguardia', 'Cultura', 'https://www.lavanguardia.com/mvc/feed/rss/cultura',false);
INSERT INTO Periodico ( nombre_periodico, tipo_informacion, link ,uso )
VALUES ( 'La Vanguardia', 'Deportes', 'https://www.lavanguardia.com/mvc/feed/rss/deportes',false);
```

Figura 54. Script SQL 4

6 Implementación

En este capítulo se explicará con profundidad, como se ha realizado la implementación de este TFG, siguiendo la arquitectura de software MVC, explicada en el capítulo anterior.

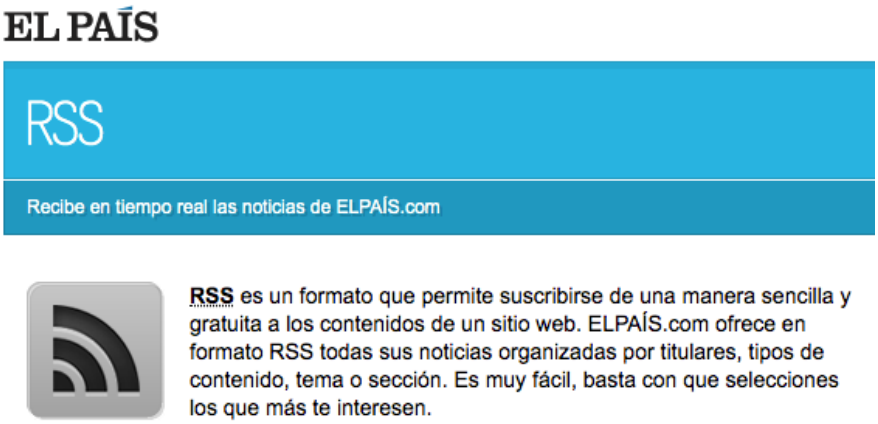
Como también se ha explicado en el apartado correspondiente las principales herramientas/tecnologías utilizadas son el lenguaje de programación Java, con el IDE NetBeans y una base de datos MySQL.

Antes de empezar, conviene explicar algunos conceptos:

6.1 RSS

RSS es la abreviación de "Really Simple Syndication", traducido al español, quiere decir "Redifusión realmente simple".
Consiste en un fichero XML con el que se comparte contenido en la Web. Esta información se encuentra actualizada y puede ser utilizada por más usuarios en otros sitios webs o programas, en lo que se conoce como redifusión web (Wikipedia, 2019).

El archivo RSS, muestra las novedades del sitio Web, como pueden ser el título, fecha de publicación o descripción. Pensar en obtener los titulares de un periódico en concreto a través de su RSS, parece lo más lógico.



(El País, 2019)

Figura 55. RSS El País

Al ser un archivo XML, supone que está compuesto por varias etiquetas definidas, cada una con un formato que respetará las normas generales de XML.

Gracias al RSS, se puede obtener de una manera fácil el contenido actualizado de una página web online, sin necesidad de tener que visitar esa página. Añadir, que la información se actualiza automáticamente, sin tener el usuario que hacer nada. El usuario sólo tiene que tener un lector de RSS.

Una de las ventajas destacables de los RSS (RSS.nom, 2019), que es la que se va a aplicar en este trabajo, es la capacidad de poder tener reunido en varios archivos, todo el contenido actualizado de varias páginas webs. Es decir, con el RSS de cada periódico, con el lector RSS, podremos estar informados de los últimos titulares publicados, así como su hora de publicación, entre otros parámetros como podría ser el autor del mismo.
De esta forma, se configurará un lector RSS, que consulte los ficheros proporcionados por varias páginas webs de periódicos de prensa, y se obtendrán los últimos titulares publicados, sin tener que consultar esas páginas Webs.
Además, nos permite conocer el horario y fecha de publicación de ese titular, dato que no aparece directamente en la página Web.

6.2 Redifusión Web

La sindicación Web o redifusión web consiste en la redistribución de contenido web, por parte de un sitio web de origen a otro usuario receptor. Para nuestro caso, las páginas de periódicos de prensa online nos facilitan su contenido informativo para la creación de este TFG.

6.3 XML

XML (Wikipedia, 2019) es la abreviación de las siglas en inglés de "eXtensible Markup Language", traducido a español significa "Lenguaje de Marcas Extensible".

Se trata de un metalenguaje, desarrollado por World Wide Web Consortium (W3C), que se usa para almacenar datos de una forma legible.

Los ficheros XML tienen muchas aplicaciones posibles como su uso en editores de texto, hojas de cálculo o bases de datos.

Se trata de una tecnología simple que se puede complementar con otras, como puede ser RSS, lo que hace que tenga un papel destacable en la actualidad, en cuanto a compartición de la información de manera segura y sencilla.

XML es un estándar internacionalmente conocido. No pertenece a ninguna compañía

Alguna de las ventajas de XML (Mundolinux, 2019), es que es extensible, como indican sus siglas, ya que se pueden añadir nuevas etiquetas. Es fácilmente procesable. Diseñado para cualquier lenguaje y alfabeto. Es sencillo de entender su estructura y procesarla.

La información aparece estructurada, con partes bien definidas por etiquetas, y que esas partes a su vez pueden estar compuestas por otras partes, con sus correspondientes etiquetas. Se tiene un árbol con la información dividida en partes.

Una etiqueta es una de las marcas, de las que se compone el documento. Muestra una porción del mismo como un elemento. Una división de la información con un sentido determinado. Las etiquetas tienen la forma de `<nombre>`.

Componentes de un documento XML:

- Elementos: Se representa con una cadena de texto, que es el dato o contenido. Aparece entre etiquetas. Pueden existir elementos vacíos `(</br>)`. Los elementos pueden contener atributos que son una manera de incorporar características o propiedades a los elementos de un documento. Deben ir entre comillas.
- Instrucciones: Órdenes especiales.
  Ejemplo: `<?xml-stylesheet type="text/css" href="estilo.css">`
- Instrucciones XML: Comienzan por "<¿" y terminan por "¿>"
- Comentarios: Comienzan por `<!-->` y terminan por `<-->`
- Declaraciones de tipo: Especifican información acerca del documento:
  Ejemplo: `<!DOCTYPE persona SYSTEM "persona.dtd">`
- Secciones CDATA: Conjunto de caracteres especiales que no tienen que ser interpretados por el procesador:
  Ejemplo: `<![CDATA[ Aquí se puede meter cualquier carácter, como <, &, >, ... Sin que sean interpretados como marcación]]>`

A continuación, es muestra un ejemplo de un archivo, donde se ve la estructura que tiene un documento XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">
```

```
<Edit_Mensaje>
     <Mensaje>
          <Remitente>
               <Nombre>Nombre del remitente</Nombre>
               <Mail> Correo del remitente </Mail>
          </Remitente>
          <Destinatario>
               <Nombre>Nombre del destinatario</Nombre>
               <Mail>Correo del destinatario</Mail>
          </Destinatario>
          <Texto>
               <Asunto>
                    Este es mi documento con una estructura muy sencilla
                    no contiene atributos ni entidades...
               </Asunto>
               <Parrafo>
                    Este es mi documento con una estructura muy sencilla
                    no contiene atributos ni entidades...
               </Parrafo>
          </Texto>
     </Mensaje>
</Edit_Mensaje>
```

Se puede apreciar como aparecen los siguientes elementos: Mensaje, Remitente, Nombre, Mail, Destinatario…

Cada una de estas etiquetas son partes de la información que guarda este documento, cada una con su sentido claro y definido.

Para nuestro caso, podemos ver el siguiente ejemplo de cómo sería el archivo XML que se leerá:

```
▼<item>
  ▼<title>
     Aguado (Cs) y Monasterio (Vox) se han reunido hoy en un hotel madrileño
   </title>
  ▼<link>
     https://www.abc.es/espana/madrid/abci-aguado-y-monasterio-reunido-hotel-madrileno-201906091744_noticia.html
   </link>
  ▼<description>
     <img align="left" src="https://www.abc.es/media/espana/2019/06/09/ignacio-aguado-madrid-kr0B--620x349@abc.jpg"
     Ciudadanos, <a href="/espana/madrid/abci-eleccoines-autonomicas-ignacio-aguado-impedire-cedan-espacios-publico
     201905140042_noticia.html" target="_blank">Ignacio Aguado,</a> y la candidata de Vox a la Comunidad de Madrid,
     domingo en un hotel madrileño</strong>. Ha sido un primer encuentro, una toma de contacto entre ambos político
     se constituya la Asamblea de Madrid y se elija al presidente de la Cámara madrileña, la primera meta que se ha
     de Ciudadanos confirmaron a ABC esta reunión, que se ha producido de forma muy discreta. No importaba la foto:
     de Vox, como ya señalaron hace días, les basta con<strong> «la foto de Colón</strong>»: ahora prefieren encuen
     concretas.</p> <p>Esta cita va a ser muy importante para que ambas formaciones lleguen a acuerdos de cara a la
     Cs quiere presidir pero para ello necesita contar con los votos de Vox. Esta formación ya había adelantado que
     antes un encuentro </strong>cara a cara, de igual a igual entre los partidos.</p> <p>Vox, de hecho, ya ha mant
     con esta de Ciudadanos, se completan sus objetivos. </p>
   </description>
   <category domain="Todos">Todos</category>
   <pubDate>Sun, 09 Jun 2019 17:44:36 +0200</pubDate>
  ▼<guid>
     https://www.abc.es/espana/madrid/abci-aguado-y-monasterio-reunido-hotel-madrileno-201906091744_noticia.html
   </guid>
   <dc:creator>(Sara Medialdea)</dc:creator>
 </item>
```

(ABC, 2019)

Figura 56. Ejemplo de XML de ABC

Cada titular aparece separado de los demás, en la etiqueta `<item>`, es decir, cada ítem será un titular diferente. Dentro del ítem, nos encontramos las siguientes partes:

`<title>` Titular
`<link>` URL de la fuente
`<description>` Información de la noticia
`<category>` Categoría de la noticia. Tipo de información

<pubDate> Fecha y horario de publicación del titular
<guid> URL de la fuente
<dc:creator> Nombre de la persona que ha publicado la noticia

Esto es un ejemplo de un titular obtenido de la fuente ABC.com
No todos los periódicos siguen esta estructura, ni dividen el titular en las mismas partes o las llaman de la misma forma, aunque hay varias similitudes.

Por ejemplo, en el RSS de el periódico El País, a mayores de las etiquetas anteriores que mostraba el periódico ABC se muestran las siguientes:
<content_encoded> Contenido codificado
<enclosure> URL de las imágenes de la noticia
<comments> URL de la fuente

```
  </category>
▼<category>
    <![CDATA[ Organizaciones deportivas ]]>
  </category>
▼<category>
    <![CDATA[ Deportes ]]>
  </category>
▼<category>
    <![CDATA[ Europa ]]>
  </category>
▼<content:encoded>
  ▼<![CDATA[
      <p>Llega mayo, y como si de una cuestión litúrgica se tratase, el pere
      </strong> a París concluye una vez más como el año anterior: elevando
      arcilla hacia el infinito. Nadal y Roland Garros, el maravilloso cuent
      encima de Novak Djokovic (15). Son ya 12 trofeos en la ciudad de la lu
      en el trazado que la planteada por el teórico sucesor, Dominic Thiem,
      href="https://elpais.com/deportes/2019/06/09/actualidad/1560075141_782
      continúa desafiando a la lógica. Parece dar igual cómo llegue, qué hay
      rebozándose de tierra.</p><p><a href="https://elpais.com/deportes/2019
      leyendo</a>.</p>
    ]]>
  </content:encoded>
  <enclosure url="https://ep00.epimg.net/deportes/imagenes/2019/06/09/actua
  type="image/jpeg"/>
▼<comments>
  ▼<![CDATA[
      http://eskup.elpais.com/C1560035497-543d52eb1b4208c0aebade781d5e9586
    ]]>
  </comments>
  <enclosure url="https://vdmedia.elpais.com/elpaistop/multimedia/20196/9/2
</item>
```

(El País, 2019)

Figura 57. Ejemplo XML El País

Hay varias etiquetas de categorías, en el País para cada titular dentro del tipo de información se consideran categorías. Como se puede apreciar un titular no pertenece a una sola categoría, sino que puede pertenecer a varias.

Hasta ahora, se ha hablado de la estructura de un documento XML. Ahora pasamos a ver sus partes. Se divide principalmente en dos: Prólogo y cuerpo.

Prólogo: Es opcional. Muestra la versión XML y el tipo de documento, así como el tipo de codificación utilizada, entre otras cosas.
Ejemplo: `<?xml version="1.0" encoding="UTF-8"?>`

Cuerpo: Es la parte obligatoria del documento XML. Se correspondería con la parte explicada con las imágenes anteriores, dónde aparecen las etiquetas.

Como se ha comentado anteriormente, se ha tomado como base de este TFG un código obtenido de Internet para la lectura de un fichero XML, a partir de una URL.

El código utilizado, como base constaba de 4 clases que procedo a explicar:

1) FeedMessage: (Anexo 1)

Clase que obtiene información de cada mensaje. Para este proyecto, se corresponde con la información del titular.

Tiene los siguientes atributos:

```
String title;
String description;
String link;
String author;
String guid;
```

Esta clase contiene Getters y Setters de los atributos y un método @Override para imprimir su contenido en un String.

2) Feed: (Anexo 2)

Clase que obtiene información de la fuente. Para este proyecto, se corresponde con la información del periódico de prensa digital.

Tiene los siguientes atributos:

```
final String title;
final String link;
final String description;
final String language;
final String copyright;
final String pubDate;
```

Esta clase contiene Getters y Setters de los atributos y un método @Override para imprimir su contenido en un String. (Arquitecturajava, 2019) y (Stackoverflow, 2019)

3) RSSFeedParser: (Anexo 3)

Se trata de la clase más importante. Es la clase que va a recibir el URL del RSS y va a leer el archivo XML.

Importa todas las 2 clases anteriores, así como las necesarias para leer un archivo XML. (Tutorials.jenkov, 2019) (Tutorials.jenkov, 2019) (Oracle, 2019) (Oracle, 2019)

```
import javax.xml.stream.XMLEventReader;
import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLStreamException;
import javax.xml.stream.events.Characters;
import javax.xml.stream.events.XMLEvent;
```

Tiene los siguientes atributos:

```
static final String TITLE = "title";
static final String DESCRIPTION = "description";
static final String CHANNEL = "channel";
static final String LANGUAGE = "language";
static final String COPYRIGHT = "copyright";
```

```
static final String LINK = "link";
static final String AUTHOR = "author";
static final String ITEM = "item";
static final String PUB_DATE = "pubDate";
static final String GUID = "guid";
final URL url;
```

Estas variables se inicializan con el texto que hay que leer del archivo XML, es decir, el nombre de las etiquetas.

Recibe como parámetro el URL del RSS para leer. Comprueba que es una URL correcta y no es errónea.

Contiene 3 métodos:

- `public Feed readFeed()`

  Guarda en unas variables el contenido que se encuentra en las etiquetas correspondientes del archivo XML. Realiza una lectura evento a evento, es decir etiqueta a etiqueta, distinguiendo entre, el evento de inicio y el evento del final.
  Crea una instancia de la clase Feed, y otra de la clase FeedMessage.
  Utiliza los Setters para dar valores, a los atributos de la clase correspondiente

- `private String getCharacterData(XMLEvent event, XMLEventReader eventReader)`

  Crea un String, carácter a carácter y devuelve el resultado.

- `private InputStream read()`

  Stream para la lectura.

4) ReadTest: (Anexo 4)
   Clase Principal, que contiene el método main().
   Importa las otras 3 clases.
   Se crear un Parser y se le proporciona el link del RSS a leer
   Se crea una instancia de la clase Feed y se impriem su contenido.
   Se crean tantas instancias como titulare haya en el archivo XML y se imprime el contenido de cada una.

Partiendo de este fragmento se ha realizado toda la implementación.

La ejecución del programa empieza con la clase Main.java. En esta clase se crea una instancia de la clase "Periodico" y otra de la clase "UsoBD". Por último, se crea la instancia de la primera vista (StartWindow) así como de su controlador (control_StartWindow).

Dentro la vista StartWindow, se muestran 3 CheckBoxes para poder seleccionar los periódicos, en concreto se pueden seleccionar:

- ABC.
- El País.
- La Vanguardia.

A mayores se muestran 5 CheckBoxes para poder seleccionar el tipo de información, son:

- Último.
- Opinión.
- Economía.
- Cultura.
- Deportes.

La información de estos periódicos se encuentra dentro de la tabla "Periódico" de la base de datos. Esta tabla se mantiene fija, no se puede modificar, es decir una vez insertadas las filas mediante el Script SQL, ya se encuentra toda la información que la herramienta utilizará.

Se decide utilizar estos tres periódicos, ya que son 3 periódicos generales, es decir, tratan varios tipos de información. Por este motivo no se ha utilizado ningún periódico deportivo, por ejemplo, como podría ser "AS" o "Marca".

Para poder realizar el procesamiento de palabras, era necesario utilizar periódicos escritos en castellano, a sabiendas, de que un titular siempre puede incluir alguna que otra palabra en otro idioma.

También hay que tener en cuenta, que no todos los RSS de los periódicos digitales poseen su información publicada por secciones tan especificas, por ejemplo, el RSS de "El día de Valladolid" no tiene sección de cultura, por lo que habría que buscar varios periódicos que tengan al menos un número mínimo de secciones iguales.

A mayores, no todos los periódicos publican su información en el fichero XML de la misma forma, ya que cambian varios nombres de etiquetas.

Teniendo en cuenta todo esto comentado, se ha decidido elegir los tres periódicos digitales ya enumerados.
Los CheckBoxes del tipo de información aparecerán deshabilitados hasta que el usuario no seleccione un periódico. Además, si el usuario deselecciona los periódicos, se deseleccionarán los CheckBoxes de tipo e información y se deshabilitarán.

Resumiendo, el usuario sólo podrá seleccionar el tipo de información, mientras tenga algún periódico seleccionado.

En la vista, aparecerá un botón "Seleccionar todos", con el que se seleccionan todos los CheckBoxes.

De forma análoga, aparecerá un botón "Eliminar selecciones", con el que se deseleccionan todos los CheckBoxes.

Con estos dos conjuntos de selecciones el usuario ya puede seleccionar qué tipo de información y de dónde quiere obtener los titulares.

El usuario también tiene la opción de seleccionar, cómo quiere que finalice el proceso de extracción de titulares. Existen dos opciones:

- Número de horas.

  El usuario especificará en el campo de texto, el número de horas que el programa se ejecutará. Para determinar el fin de la ejecución, se almacena en el modelo la hora de inicio del programa y se realiza la suma de horas que el usuario ha especificado. De esta forma, también en el modelo se almacena la fecha de finalización de ejecución.

  Cada vez que se ejecute el código periódicamente se hace una comparación de la hora actual con la hora establecida para la finalización. Por tanto, si se supera la hora, el proceso de extracción de titulares finaliza, y se muestra el resumen de resultados.

- Número de titulares.

  El usuario especificará en el campo de texto, el número de titulares que el programa extraerá. Para ver el número de titulares al igual que el método anterior se guarda la fecha de inicio del programa y se hace una comparación con las horas de publicación de los titulares.

  Por lo tanto, cuando se hayan almacenado el número de titulares especificado, con una fecha de publicación superior a la de la hora de inicio, el proceso de extracción finaliza.

  Conviene aclarar que no siempre que se almacena un titular nuevo, se trata de un titular con una fecha de publicación posterior a la fecha de inicio. Esto se debe a que, en el RSS, a veces, no siempre, se publican titulares con una fecha de publicación anterior, por ejemplo, de hace una hora, por tanto, ese titular no contaría como titulares almacenados a partir de la fecha de inicio.

  También se dan casos del caso contrario, se publican titulares con una fecha de publicación que todavía no ha llegado, por ejemplo, un titular con una fecha de publicación de dentro de 5 horas.

Una vez que el usuario ha seleccionado toda la información, pulsará el botón de Inicio.
Es necesario que el usuario especifique todos los campos, es decir:

1) Al menos un periódico.
2) Al menos un tipo de información.
3) Seleccionado tipo de finalización del programa.
4) Especificado el número de horas o número de titulares.

Si alguna de estas condiciones no se cumple, el programa mostrará un mensaje de error, indicándole al usuario qué condición no cumple, antes de dejarle continuar.

Al pulsar el botón de Inicio, con todas las condiciones satisfechas, se mostrará la siguiente vista, la vista Results.java con los gráficos y tablas de los datos almacenados en la base de datos.

En este punto, se realiza la conexión con la base de datos. La clase encargada de realizar dicha función se denomina Conexión.java (Anexo 5) y se encuentra dentro del Modelo.

Es necesario añadir el siguiente archivo JAR al proyecto: mysql-connector-java-5.1.47

User mysql: root
Password: bicicleta
Nombre de base de datos: tfg

Se crean dos métodos:
1) hazConexion() Conectarse con la base de datos
2) desconectar() Desconectarse de la base de datos

La conexión se realiza con la siguiente línea:

```
conectar =(Connection)DriverManager.getConnection("jdbc:mysql://
localhost:3306/tfg?autoReconnect=true&useSSL=false", "root","bicicleta");
```

Se especifica que la base de datos es local. Hay que proporcionar el nombre de la base de datos, así como el usuario y contraseña del usuario de MySQL, donde se encuentre dicha base de datos.

Cada vez que se necesite acceder a la base de datos ya sea para insertar o eliminar una fila, se creará una conexión con la misma. Se ejecutarán las ordenes correspondientes y se cerrará la conexión.

Lo mismo ocurre, cuando se quiera realizar una consulta a la misma.

Al pulsar en el botón de Inicio, se actualiza la tabla Periódico en función de la selección de periódicos y de tipo de información que el usuario haya realizado para dicho experimento. Es necesario actualizar el campo uso de la tabla Periodico en función de la elección del usuario:

```
UPDATE Periodico SET uso=? WHERE nombre_periodico=? AND tipo_informacion=?
```

Se marca el campo de uso a true de las URLs que el usuario ha decidido utilizar.
Una vez, que la base de datos tiene la información de los enlaces que hay que consultar durante el proceso de extracción, se comienza la ejecución repetida del código. Para ello, se utiliza un Timer de Java que se procede a explicar.

Para la realización de este TFG, surge la necesidad de realizar una tarea cada cierto tiempo, es decir, realizar una tarea periódicamente. Se trata de consultar los RSS de las fuentes de los periódicos de prensa online con el fin de encontrar los nuevos titulares publicados.

Para solventar, esta necesidad me ayudé de las clase: java.util.Timer y java.util.TimerTask

Resumiendo, la combinación de estas clases permite llamar a un método implementado, cada un cierto tiempo especificado en milisegundos.

A continuación, se muestra un ejemplo de cómo funciona esta combinación de clases.
Dentro del método run(), se encuentra el código necesario para la extracción de los titulares de prensa.

```
// Clase en la que está el código a ejecutar
TimerTask timerTask = new TimerTask()
{
    public void run()
    {
        // Aquí el código que queremos ejecutar.
    }
};
....

 // Aquí se pone en marcha el timer cada segundo.
Timer timer = new Timer();
// Dentro de 0 milisegundos avísame cada 1000 milisegundos
timer.scheduleAtFixedRate(timerTask, 0, 1000);
```

(Chuidiang, 2019)

Figura 58. Ejemplo Timer

El tiempo, se especifica en milisegundos. Para este proyecto, un intervalo razonable para ejecutar el código es cada 30 minutos. Esto se debe a que, los periódicos no publican titulares constantemente. De la publicación del ultimo titular al siguiente pasa un tiempo variable de tiempo. Puede ser corto, de un minuto o menos, o pasar una hora sin que se publique ningún titular. Este último caso se da por las noches.

Hay que tener en cuenta, que tampoco se puede poner un periodo de tiempo muy grande, ya que se podría correr el riesgo de perder algún titular. Este caso puede darse cuando, si se ejecuta el código por ejemplo cada 2 horas o más, y durante esas 2 horas se publican 25 titulares nuevos.

Como se ha comentado ya en este documento, en el RSS de la fuente sólo se encuentran un número limitado de titulares, que se corresponden con los últimos publicados, por lo que, de una ejecución a otra, en un largo periodo de tiempo se ha podido renovar completamente ese archivo.

Visto esto, se ha considerado que un intervalo de 30 minutos es un periodo razonable, para evitar este problema,

El Timer se ha especificado en la clase ReadTest.java, donde se repite periódicamente el método run(). Lo primero de todo es obtener la información de la tabla Periódico de la base de datos, donde aparecerán con el campo en true, de aquellos links que hay que consultar.

Para cada una las URLs se realiza una conexión a Internet para acceder a su archivo XML. Puede darse el caso de que haya un problema con la conexión, provocado por un fallo de red del usuario o por parte de la página web.

Si en algún momento esto ocurriera y el programa no pudiera acceder a una determinada dirección, la obviaría por el momento y pasaría ala siguiente. En la próxima iteración se volverá a intentar acceder a ella.

Si no hay ningún error se guardan en un ArrayList de mensajes todos los titulares, junto con sus campos.

Para añadir un nuevo titular a la base de datos, previamente se ha tenido que comprobar que ese titular no estaba ya añadido.

Para cada uno de los titulares que se encuentran el ArrayList se realiza el siguiente proceso que. consta de dos pasos: el primero de una consulta y el segundo de una inserción:

1)Búsqueda.

Consiste en un método booleano que dependiendo de si el titular está guardado devuelve un true y false en el caso contrario.
Esta consulta se realiza para cada uno de los titulares leídos del RSS.

```
SELECT *
FROM Titulares
WHERE titulo =?;
```

2)Inserción.

Si se ha comprobado que el titular no está almacenado, es decir, el primer paso nos ha devuelto un false, se puede proceder a la creación de una nueva observación de la tabla Titulares, con los valores correspondientes para cada una de las columnas.

```
  INSERT INTO Titulares (titulo, titulo_limpio, dia_semana,
  dia, mes, hora, palabras, caracteres, nombre_periodico, tipo_informacion,
num) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?,?)
```

Este mismo proceso de dos pasos se realiza con la tabla de Palabras, con una pequeña diferencia, ya que para cada palabra se almacena el identificador del titular en el que aparece, por tanto, se almacenan varias apalabras iguales, que han aparecido por tanto en varios titulares. Puede darse el caso, que, en un mismo titular, una misma palabra aparezca repetida, por lo que se incluye un campo contador, que muestra la cantidad de veces que aparece esa palabra determinada en el titular.

1) Búsqueda.

Al igual que para el caso de los titulares, consiste en un método booleano que devuelve true si la palabra de un mismo titular ya está almacenada o false si no lo está. Es decir, comprueba si en un mismo titular, la palabra aparece repetida.

Se realiza únicamente con las palabras de los titulares nuevos, es decir, aquellos que van a ser insertados en la base de datos. No con cada una de las palabras leídas del RSS.

```
SELECT *
FROM Palabras
WHERE palabra =? AND nombre_periodico= =?
AND tipo_informacion = ? AND num =?;
```

2) Inserción

Si la palabra está repetida en el titular, se realiza una actualización de dicha fila de la base de datos, modificando su contador al incrementarlo en una unidad.

```
UPDATE Palabras SET contador=? WHERE palabra= ?
AND nombre_periodico=? AND tipo_informacion=?
```

Si la palabra no está repetida, se inserta en la base de datos, indicando el identificador del titular al que pertenece.

```
INSERT INTO Palabras(palabra,nombre_periodico,tipo_informacion,
contador,num,negativa,busqueda) VALUES (?, ?, ?, ?, ?, ?, ?)
```

Con este proceso, se han comprobado todos los titulares que aparecen en cada una de los ficheros XML. Este proceso se repite cada 30 minutos con la finalidad de extraer los nuevos titulares que han sido publicados durante este tiempo.

Se realiza una consulta a la base de datos para comprobar si el titular ya ha sido almacenado. Esto mismo se podría realizar con una sola consulta que guarde en un ArrayList todos los titulares de la base de datos, y se busque en ese ArrayList cada uno de los titulares leídos del RSS.

Esta última opción, pese a que sólo realiza una consulta a la base de datos, puesto que para cada titular leído de internet habría que recorrer el ArrayList que puede contener miles de titulares.

Cuando se acaban de leer todas las URLs especificas se realiza la comparación para ver si el programa tiene que finalizar. Dependiendo de la opción elegida se evaluará si se ha cumplido el tiempo d ejecución o si ya se han almacenado el número de titulares proporcionado.

Mientras se realiza el proceso de extracción el usuario se encontrará en la vista de Results, donde podrá ver en directo los gráficos actualizados con los datos que se encuentran en la base de datos. Por cada titular que se añade se modifica la vista actualizándose los gráficos y tabla.

Conviene decir, que en esta vista el usuario tiene la capacidad de seleccionar de qué periódico de los tres disponibles, quiere ver los gráficos. Sin embargo, mientras se está actualizando la base de datos, el usuario no puede modificar esa selección. El programa indicará al usuario cuando se está actualizando la base de datos y cuándo puede consultar los gráficos modificando las selecciones.

A continuación, se procede a explicar en qué consiste el proceso de limpieza de titulares:
Durante el proceso de extracción de los titulares se realiza un proceso de limpieza de los mismos, es decir, eliminar aquellas palabras que no se consideran relevantes.

Se realiza paso a paso eliminado aquellas partes del titular que no interesan: Se podrían resumir en:

1) Quitar signos:
   Creación de una lista de palabras con todos los signos de ortografía, interrogación, exclamación…, para su posterior eliminación.

2) Quitar preposiciones
   Creación de una lista de palabras con todas las preposiciones, para su posterior eliminación.

3) Quitar artículos:
   Creación de una lista de palabras con los artículos gramaticales, para su posterior eliminación.

4) Quitar conjunciones
   Creación de una lista de palabras con las conjunciones gramaticales, para su posterior eliminación.

5) Quitar otras palabras
   Creación de una lista de palabras formada por aquellas palabras, como pueden ser algunos determinantes posesivos, o verbos auxiliares entre otras, para su posterior eliminación.

A continuación, se explica cómo se obtienen los campos de la tabla de Titulares:

La tabla Titulares, tiene las siguientes columnas:

1) Titulo: Obtenido directamente del archivo XML del RSS.

2) Titulo limpio: Título tras el proceso de eliminación de las palabras con poca relevancia.

Los cuatro próximos campos se obtienen del contenido de la etiqueta pubDate del archivo XML del RSS. Un ejemplo de lo que contiene dicho campo es el siguiente: Mon, 1 Apr 2019 17:47:33 +0200.
Dividiéndolo por campos separados por espacios se obtienen los valores que almacenamos.

3) Día semana: Primer campo

4) Día: Segundo campo

5) Mes: Tercer campo

6) Hora: Quinto campo
7) Palabras: Número de palabras que contiene el título completo, todavía sin limpiar.

8) Caracteres: Número de caracteres que contiene el título completo, todavía sin limpiar.

9) Nombre periódico: Nombre del periódico, al que pertenece dicho titular que se obtiene de la tabla Periódico.

10) Tipo información: Tipo de información al que pertenece dicho titular que se obtiene de la tabla Periódico.

11) Num. Sirve de identificador, el primer titular almacenado es el 1 y a medida que se van almacenando titulares, se incrementa su valor. Sirve también para determinar el número de titulares almacenados.

A continuación, se explica cómo se obtienen los campos de la tabla de Titulares:

La tabla Palabras, tiene las siguientes columnas:

1) Palabra: Cada una de las palabras que forman el titular limpio.

2) Nombre Periódico: Nombre del periódico, donde ha sido publicado el titular al que pertenece la palabra, se obtiene de la tabla Periódico.

3) Tipo Información: Tipo de información, del titular al que pertenece la palabra, se obtiene de la tabla Periódico.

4) Contador: Veces que aparece la palabra en el titular.

5) Num: Identificador del titular al que pertenece la palabra.

6) Negativa: Booleano que indica si la palabra está considerada dentro de la lista de palabras que se han especificado como negativas.

7) Búsqueda: Booleano que indica si la palabra ha sido especificada como palabra a tener en cuenta en la búsqueda de titulares.

Cuando ha finalizado el proceso de extracción de titulares, el usuario se encontrará en la vista de Final_Results.java, donde el usuario podrá ver un resumen del número de titulares almacenados en la base de datos. También tiene la opción de visualizar otra vez los gráficos.

Se plantea la posibilidad, de que el usuario pueda proporcionar una lista de palabras positivas y otra lista de palabras negativas. La finalidad de esto, es poder identificar titulares positivos y titulares negativos. Está claro que es muy difícil poder conocer la verdadera intencionalidad o sensibilidad de un titular.
Simplemente con esto, podemos hacer una clasificación entre los titulares de si son positivos, negativos y neutrales.

Si en el titular aparece al menos una palabra negativa, se considera un titular negativo.
Si en el titular aparece al menos una palabra positiva, se considera un titular positivo.
Puede darse el caso de que aparezca en el titular tanto una palabra positiva como una negativa, en ese caso se contará el número de palabras positivas y negativas, Si el numero coincide será considerado neutral.
Como cabe esperar, si el titular no aparece ninguna palabra positiva ni negativa, también será considerado como un titular neutral.

En esta vista el usuario puede realizar las siguientes acciones:

- Importar palabras negativas.
- Importar palabras positivas.
- Importar palabras de búsqueda.
- Procesar palabras.

El proceso de importar palabras, ya sean positivas, negativas o de búsqueda se realiza de la misma forma.

El usuario dispondrá de tres archivos CSV, uno para cada tipo de palabras.
Estos archivos se deberán encontrar en la ruta especificada en el manual. Los archivos contendrán una lista de palabras que el usuario tendrá la capacidad de modificar cada uno de los archivos, eliminando o añadiendo las palabras deseadas. En cada archivo se mostrará una palabra por línea.

Una vez que el usuario, ha modificado los archivos si es que asó lo ha deseado, en la vista del programa podrá pulsar los botones de importar palabras. De esta forma, se actualizará la tabla de la base de datos correspondientes a cada una de las tres listas de palabras.

Se mostrará un mensaje al usuario indicando que el proceso de importar las palabras ya actualización de la base de datos se ha realizado correctamente.

Por último, una vez que se han importado las palabras, el usuario puede Procesar las palabras con su botón correspondiente. Con esto se actualizarán los campos indicando si cada palabra es positiva, negativa o de búsqueda, comparando la lista de palabras totales almacenadas en la base de datos con cada una de las 3 listas de palabras.

Al igual que al importar las palabras, se mostrará un mensaje al usuario indicando que el procesamiento de palabras se ha realizado correctamente.

De esta forma, realizando el procesamiento de palabras al final de la extracción de titulares, da la capacidad al usuario de introducir palabras que no tenía pensado buscar antes de la extracción.

Si el usuario, especificara las palabras antes del proceso de extracción, el procesamiento sería mucho menos flexible de como lo es ahora. El programa gana potencia y flexibilidad para las necesidades del usuario.

A la hora de determinar si una palabra es negativa o positiva, se ha elaborado una lista, con alrededor de 50 palabras. Sin embargo, para que la comparación de las palabras no se realice exclusivamente con esa lista, se utiliza un método para determinar la similitud entre palabras. Si la similitud es grande, se considerará también como una palabra negativa.
Este método, supone que se amplíen las posibilidades de encontrar una palabra negativa, sin embargo, se corre un riesgo, y es el considerar palabras como negativas cuando no lo son, por el simple hecho de ser muy parecida a una de la lista. Esto es un error asumible.

Para determinar la similitud entre palabras se ha utilizado la distancia de Levenshtein (Wikipedia, 2019).

La distancia de Levenshtein, también conocida como distancia entre palabras o distancia de edición, se considera el mínimo número de operaciones que se necesitan para transformar una palabra o cadena de caracteres en otra.
Existen tres tipos de operaciones que son: inserción, eliminación y sustitución de un carácter.

Generaliza la distancia de Hamming que sólo es utilizada para palabras con la misma longitud y que sólo considera la operación de sustitución de un carácter.

Para cada palabra del titular ya limpio, se calcula la afinidad mediante la distancia de Levenshtein (Censorcosmico, 2019)con las palabras de la lista creada a mano. Si la afinidad es mayor de un 75% se considera negativa, y si no lo es, no se considera negativa.

Evidentemente si la palabra es una incluida, dentro de la lista, la afinidad será del 100% por lo que se añadirá como palabra negativa igualmente.

Con esto se darán varios problemas, que se consideran asumibles y entendibles.

1) Considerar una palabra negativa no siéndolo realmente.
2) No considerar una palabra negativa, siéndolo realmente.

Conviene destacar que el valor del 75%, se puede modificar en el código fuente, y establecerlo con mayor o menor afinidad.

Una vez que se ha realizado el procesamiento de palabras, el usuario tiene la posibilidad de ver aquellos titulares que se han considerado negativos o positivos. Además, también puede ver aquellos titulares que contienen una palabra que el usuario ha considerado de búsqueda.

Pulsando en el botón ver titulares, el programa mostrará al usuario una tabla con los titulares. El usuario tendrá la capacidad de seleccionar que titulares quieren ver, si los positivos, los negativos, o aquellos que contienen alguna palabra buscada.

En este punto la ejecución del programa se podría considerar finalizada. El usuario podría utilizar otros programas para aprovechar los datos almacenados en la base de datos.

Conectando estos programas de visualización de datos como son RStudio o Power BI con la base de datos, las posibilidades de representación y visualización de datos son enormes.

7 Pruebas

En este capítulo se muestran todas aquellas pruebas realizadas sobre la herramienta desarrollada. Esas pruebas tienen la finalidad de certificar la fiabilidad de aquellas funcionalidades que presenta la herramienta, así como la detección de la gran parte de defectos.

Las pruebas se han realizado a medida que el programa se ha ido desarrollando, periódicamente con la introducción de una nueva funcionalidad o modificación de una existente.

La metodología empleada ha sido de caja negra, basándose en los requisitos y casos de uso especificados en su capítulo correspondiente.

A continuación, se muestra un resumen de las pruebas realizadas, con los fallos más significativos e importantes.

Tabla 15. Prueba 1

Prueba 1	
Descripción	El usuario inicia el programa.
Resultado esperado	La vista para seleccionar el tipo de información es mostrada.
Resultado	Resultado esperado.

Tabla 16. Prueba 2

Prueba 2	
Descripción	El usuario selecciona uno o más periódicos con los CheckBoxes correspondientes.
Resultado esperado	Se habilitan las CheckBoxes para poder seleccionar el tipo de información.
Resultado	Resultado esperado.

Tabla 17. Prueba 3

Prueba 3	
Descripción	El usuario deselecciona todos los periódicos que tenía marcados en los CheckBoxes correspondientes.
Resultado esperado	Se deshabilitan y se deseleccionan las CheckBoxes para poder seleccionar el tipo de información.
Resultado	Resultado esperado.

Tabla 18. Prueba 4

Prueba 4	
Descripción	El usuario no puede comenzar el proceso de extracción de titulares sin al menos haber elegido un periódico.
Resultado esperado	Un mensaje de error será mostrado al usuario indicando que debe introducir al menos un periódico.
Resultado	Resultado esperado.

Tabla 19. Prueba 5

Prueba 5	
Descripción	El usuario no puede comenzar el proceso de extracción de titulares sin al menos haber elegido un tipo de información.
Resultado esperado	Un mensaje de error será mostrado al usuario indicando que debe introducir al menos un tipo de información.
Resultado	Resultado esperado.

Tabla 20. Prueba 6

Prueba 6	
Descripción	El usuario no puede comenzar el proceso de extracción de titulares sin haber elegido un tipo de finalización, ya sea por horas o por número de titulares.
Resultado esperado	Un mensaje de error será mostrado al usuario indicando que debe seleccionar un método de finalización.
Resultado	Resultado esperado.

Tabla 21. Prueba 7

Prueba 7	
Descripción	El usuario no puede comenzar el proceso de extracción de titulares sin haber introducido el número de horas o el número de titulares, para la finalización del proceso.
Resultado esperado	Un mensaje de error será mostrado al usuario indicando que debe introducir el número de horas o el número de titulares, para la finalización del proceso.
Resultado	Resultado esperado.

Tabla 22. Prueba 8

Prueba 8	
Descripción	El usuario puede seleccionar todos los tipos de información pulsando el botón “Seleccionar todos”.
Resultado esperado	Todos los CheckBoxes de tipos de información se seleccionan.
Resultado	Resultado esperado.

Tabla 23. Prueba 9

Prueba 9	
Descripción	El usuario puede deseleccionar todos los periódicos y tipos de información pulsando el botón “Eliminar selecciones”.
Resultado esperado	Todos los CheckBoxes de periódicos y tipos de información se deseleccionan.
Resultado	Resultado esperado.

Tabla 24. Prueba 10

Prueba 10	
Descripción	El usuario puede comenzar el proceso de extracción de titulares.
Resultado esperado	La vista con los resultados en directos del proceso de extracción es mostrada, con un dashboard formado por 1 tabla y 3 gráficos.
Resultado	Resultado esperado.

Tabla 25. Prueba 11

Prueba 11	
Descripción	Tabla de resultados.
Resultado esperado	El usuario puede ver una tabla en la parte superior izquierda, mostrando la estadística seleccionada para el periódico seleccionado.
Resultado	Resultado esperado.

Tabla 26. Prueba 11

Prueba 11	
Descripción	Gráfico Pie Chart con número de titulares por tipo de información.
Resultado esperado	El usuario puede ver un gráfico Pie Chart en la parte superior derecha, mostrando a proporción del número de titulares por cada tipo de información para el periódico seleccionado.
Resultado	Resultado esperado.

Tabla 27. Prueba 12

Prueba 12	
Descripción	Gráfico de barras con el número medio de palabras por tipo de información.
Resultado esperado	El usuario puede ver un gráfico de barras horizontales en la parte inferior izquierda, mostrando el número promedio de palabras de los titulares de cada tipo de información para el periódico seleccionado.
Resultado	Resultado esperado.

Tabla 28. Prueba 13

Prueba 13	
Descripción	Gráfico de barras con el número medio de caracteres por tipo de información.
Resultado esperado	El usuario puede ver un gráfico de barras horizontales en la parte inferior derecha, mostrando el número promedio de caracteres de los titulares de cada tipo de información para el periódico seleccionado.
Resultado	Resultado esperado.

Tabla 29. Prueba 14

Prueba 14	
Descripción	El usuario puede seleccionar otro tipo de estadísticas mediante los radio buttons correspondientes.
Resultado esperado	La tabla cambia mostrando las estadísticas seleccionadas para el periódico correspondiente.
Resultado	Resultado esperado.

Tabla 30. Prueba 15

Prueba 15	
Descripción	El usuario puede seleccionar otro tipo de periódico mediante los radio buttons correspondientes.
Resultado esperado	Tanto la tabla, como el gráfico Pie Chart, como los dos gráficos de barras horizontales se cambian mostrando las estadísticas para el periódico correspondiente.
Resultado	Resultado esperado.

Tabla 31. Prueba 16

Prueba 16	
Descripción	Barra de progreso.
Resultado esperado	El usuario puede ver en la parte superior de la pantalla de la vista de resultados en directo, una barra de progreso que mostrará al usuario la proporción de tanto el tiempo que resta como el número de titulares que faltan hasta la finalización del proceso.
Resultado	Resultado esperado.

Tabla 32. Prueba 17

Prueba 17	
Descripción	Actualización de barra de progreso.
Resultado esperado	Por cada titular añadido o a medida que pasa el tiempo, la barra de progreso se actualiza.
Resultado	Resultado esperado.

Tabla 33. Prueba 18

Prueba 18	
Descripción	Actualización de tabla y gráficos.
Resultado esperado	Cuando se añade un nuevo titular se actualizan los gráficos y la tabla, mostrando las nuevas estadísticas actualizadas.
Resultado	Resultado no esperado. Cuando se almacenaba un nuevo titular no se actualiza la tabla ni los gráficos, se queda estático.
Acción	Cuando se añade un nuevo titular, hay que actualizar una consulta a la base de datos para poder actualizar la tabla y gráfico correspondiente.

Tabla 34. Prueba 19

Prueba 19	
Descripción	Interrumpir proceso de extracción.
Resultado esperado	El usuario al interrumpir el proceso de extracción saliendo del programa, verá un mensaje indicando si realmente quiere salir o prefiere seguir con el proceso de extracción de titulares.
Resultado	Resultado esperado.

Tabla 35. Prueba 20

Prueba 20	
Descripción	Último titular.
Resultado esperado	El usuario puede ver en la parte inferior de la pantalla de la vista de resultados en directo, un campo que mostrará al usuario el último titular que ha sido añadido a la base de datos, así como el periódico en el que ha sido publicado y el tipo de información que trata.
Resultado	Resultado esperado.

Tabla 36. Prueba 21

Prueba 21	
Descripción	Actualización del último titular.
Resultado esperado	El campo del ultimo titular se actualizará cuando se añada un nuevo titular a la base de datos.
Resultado	Resultado esperado.

Tabla 37. Prueba 22

Prueba 22	
Descripción	Radio Buttons deshabilitados durante el proceso de extracción.
Resultado esperado	Mientras se está actualizando la base de datos no es posible cambiar las selecciones de visualización de los gráficos, aparecerá un mensaje indicando “Actualizando la base de datos”.
Resultado	Resultado esperado.

Tabla 38. Prueba 23

Prueba 23	
Descripción	Fin proceso de extracción.
Resultado esperado	Cuando finalice el proceso de extracción, es decir cuando haya transcurrido el tiempo establecido por el usuario o se haya almacenado el número de titulares especificados por el usuario desaparecerá la vista actual.
Resultado	Resultado esperado.

Tabla 39. Prueba 24

Prueba 24	
Descripción	Número de titulares almacenados.
Resultado esperado	Se mostrará una nueva vista que indica un resumen de los titulares almacenados en la base de datos, con el número total de titulares almacenados, así como el número por periódicos.
Resultado	Resultado esperado.

Tabla 40. Prueba 25

Prueba 25	
Descripción	Ver gráficos.
Resultado esperado	En la vista resumen el usuario al pulsar en el botón “Ver Gráficos”, la vista para poder visualizar otra vez la tabla y los gráficos será mostrada.
Resultado	Resultado esperado.

Tabla 41. Prueba 26

Prueba 26	
Descripción	Volver a vista de resultados.
Resultado esperado	El usuario puede volver a la vista de resumen de resultados.
Resultado	Resultado esperado.

Tabla 42. Prueba 27

Prueba 27	
Descripción	Importar palabras negativas.
Resultado esperado	En la vista de resumen de resultados, el usuario al pulsar en el botón "Importar Negativas", se actualizará la tabla de palabras negativas de la base de datos con las proporcionadas en el archivo CSV correspondiente. Aparecerá un mensaje indicando que la importación se ha realizado.
Resultado	Resultado no esperado. Al intentar importar un archivo CSV especificando una ruta, MySQL debido a su protección de seguridad para importar archivos, no dejaba cargar el archivo.
Acción	Colocar el archivo correspondiente en la ruta considerada segura por MySQL que se trata de: C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/

Tabla 43. Prueba 28

Prueba 28	
Descripción	Importar palabras positivas.
Resultado esperado	En la vista de resumen de resultados, el usuario al pulsar en el botón "Importar Positivas", se actualizará la tabla de palabras positivas de la base de datos con las proporcionadas en el archivo CSV correspondiente. Aparecerá un mensaje indicando que la importación se ha realizado.
Resultado	Mismo resultado que la Prueba 27
Acción	Misma acción que la Prueba 27

Tabla 44. Prueba 29

Prueba 29	
Descripción	Importar palabras de búsqueda.
Resultado esperado	En la vista de resumen de resultados, el usuario al pulsar en el botón "Importar Búsqueda", se actualizará la tabla de palabras de búsqueda de la base de datos con las proporcionadas en el archivo CSV correspondiente. Aparecerá un mensaje indicando que la importación se ha realizado.
Resultado	Mismo resultado que la Prueba 27
Acción	Misma acción que la Prueba 27

Tabla 45. Prueba 30

Prueba 30	
Descripción	Procesamiento de palabras.
Resultado esperado	En la vista de resumen de resultados, el usuario al pulsar en el botón "Importar Búsqueda", se actualizará la tabla de palabras de la base de datos, con los nuevos valores de los campos para las palabras que han sido consideradas positivas, negativas y de búsqueda con la ultima importación de los archivos.
Resultado	Resultado esperado.

8 Conclusiones

8.1 Objetivos alcanzados

A lo largo de la realización de este TFG se han conseguido una serie de objetivos.

De manera global, se puede indicar que la aplicación realizada que satisface la necesidad de un usuario de conocer estadísticas de titulares de prensa, teniendo la posibilidad de realizar un estudio comparativo entre esos titulares.

De forma más detallada:

- La herramienta, consiste en un experimento donde el usuario es capaz de seleccionar la información que desea para realizar la extracción de titulares y su posterior estudio estadístico.

- La herramienta es capaz de extraer los titulares tal cual han sido publicados, procesarlos, almacenarlos en una base de datos y mostrar gráficos estadísticos.

- En el desarrollo de este TFG, he podido aprender lo importante que es realizar una buena planificación. Se puede apreciar, como la estimación inicial de duración del proyecto se ha visto incrementada un tercio.

- Ha sido la primera toma de contacto con un proyecto que se puede considerar de larga duración (4 meses), con una cantidad de trabajo más elevada que la realización de una práctica, como han sido cualquiera de las desarrolladas a lo largo del Grado.

- Ha sido necesario hacer una memoria con toda la documentación de todo lo que ha consistido este TFG. A lo largo de la carrera, las memorias se centraban básicamente en una asignatura y este caso se ha desarrollado una memoria utilizando el conocimiento adquirido en un buen número de asignaturas.

- Me ha dado la oportunidad de aprender un nuevo programa, para la visualización de datos, como es Power BI.

8.2 Trabajo futuro

Pese a que la aplicación se puede considerar cerrada, puesto que su finalidad de crear una herramienta para el posterior análisis de estadísticas con otro programa, se puede considerar satisfactoria, existen algunos puntos de mejora que se proceden a explicar.

- El procesamiento de titulares, es decir, la eliminación de palabras que se consideran menos relevantes, se realiza proporcionando una lista elaborada a mano con las palabras que se quieren eliminar.

  Una mejora sería implementar una funcionalidad que sea capaz de determinar el tipo de palabra que forma cada titular. El castellano tiene nueve diferentes tipos de palabras que son: artículos, sustantivos, pronombres, adjetivos, verbos, adverbios, preposiciones conjunciones e interjecciones.

  Con esta funcionalidad, sabiendo el tipo de palabra que es cada una, dependiendo del tipo, se eliminaría o no. Un ejemplo, sería que todas aquellas palabras consideradas artículos, preposiciones, conjunciones e interjecciones no fueran consideradas para el hipotético estudio estadístico (basado en la semántica de los titulares), pues el valor semántico generalmente no se ve grandemente afectado por estos tipos de palabras.

  Con una lista de palabras, siempre quedará alguna palabra del castellano que se haya olvidado añadir, y por tanto no será eliminada.

- Implementar un formulario para que el usuario pueda añadir nuevos periódicos y tipos de información con su correspondiente link del RSS.

- Implementar una opción para importar archivos a la base de datos fuera de la ruta considerada segura por MySQL. Una ruta más accesible para el usuario.

9 Bibliografía

ABC. (2019). *Captura RSS ABC*. Recuperado el 27 de Mayo de 2019, de https://www.abc.es/rss/feeds/abcPortada.xml

Arquitecturajava. (2019). *Override*. Recuperado el 3 de Junio de 2019, de https://www.arquitecturajava.com/java-override-y-encapsulacion/

Astah. (2019). *Logo Astah UML*. Recuperado el 27 de Mayo de 2019, de http://astah.net/

Censorcosmico. (2019). *Implementación distancia de Levenshtein*. Recuperado el 10 de Junio de 2019, de https://censorcosmico.blogspot.com/2016/11/calculo-del-grado-de-similitud-o.html

Chuidiang. (2019). *Captura Ejemplo Timer*. Obtenido de http://www.chuidiang.org/java/timer/timer.php

Chuidiang. (2019). *Ejemplos Timer Java*. Recuperado el 3 de Junio de 2019, de http://www.chuidiang.org/java/timer/timer.php

Codingornot. (2019). *Esquema MVC*. Recuperado el 27 de Mayo de 2019, de https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve

Cotterell, B. H. (2002). *Software Project Management*. McGraw Hill. Obtenido de Bob Hughes and Mike Cotterell. Software Project Management. McGraw Hill, 2002.

Dennysjymbo. (2019). *XML Parser*. Recuperado el 3 de Junio de 2019, de http://dennysjymbo.blogspot.com/

Dri. (2019). *Logo MySQL*. Recuperado el 27 de Mayo de 2019, de https://dri.es/album/blog/mysql-logo

El País. (2016). *Era digital*. Recuperado el 6 de Mayo de 2019, de El País: https://elpais.com/elpais/2016/04/13/opinion/1460540302_620130.html

El País. (2019). *Captura El Pais RSS*. Recuperado el 27 de Mayo de 2019, de https://servicios.elpais.com/rss/

El País. (2019). *Captura RSS El País*. Recuperado el 27 de Mayo de 2019, de http://ep00.epimg.net/rss/tags/ultimas_noticias.xml

Elfuturodelosdatos. (2019). *Power BI*. Recuperado el 3 de Junio de 2019, de https://elfuturodelosdatos.com/power-bi-tutorial-espanol/

GitLab. (2019). Obtenido de https://gitlab.inf.uva.es

Isaacfigueroa. (2019). *Logo Power BI*. Recuperado el 27 de Mayo de 2019, de https://isaacfigueroa.com/podcast/49-juntando-informacion-power-bi-desktop/

Java RSS Feed. (2019). *Java RSS Feed*. Recuperado el 3 de Junio de 2019, de https://www.vogella.com/tutorials/RSSFeed/article.html

Macupdate. (2019). *Logo MySQL WorkBench*. Recuperado el 27 de Mayo de 2019, de https://www.macupdate.com/app/mac/31829/mysql-workbench

Maximaformacion. (2019). *Logo R*. Recuperado el 27 de Mayo de 2019, de https://www.maximaformacion.es/blog-dat/que-es-r-software/

Microsoft. (2019). *Power BI*. Recuperado el 3 de Junio de 2019, de https://docs.microsoft.com/es-es/power-bi/desktop-getting-started

Microsoft. (2019). *Power BI Desktop*. Recuperado el 20 de Mayo de 2019, de https://docs.microsoft.com/es-es/power-bi/desktop-getting-started

Mundolinux. (2019). *XML*. Recuperado el 20 de Mayo de 2019, de http://www.mundolinux.info/que-es-xml.htm

Oracle. (2019). *InputStream*. Recuperado el 3 de Junio de 2019, de https://docs.oracle.com/javase/7/docs/api/java/io/InputStream.html

Oracle. (2019). *XMLEvent*. Recuperado el 3 de Junio de 2019, de https://docs.oracle.com/javase/8/docs/api/index.html?javax/xml/stream/events/XMLEvent.html

Palkotools. (2011). *Importar datos RSS*. Recuperado el 3 de Junio de 2019, de http://palkotools.blogspot.com/2011/06/tutorial-how-to-import-rss-feeds-into.html

PDSC. (2019). *Gestión de Riesgos*. Recuperado el 20 de Mayo de 2019, de https://aulas.inf.uva.es/pluginfile.php/44263/mod_resource/content/5/PyDSC_riesgos_1819.pdf

Poesiabinaria. (2019). *Timer Java*. Recuperado el 3 de Junio de 2019, de https://poesiabinaria.net/2014/01/intro-timertask-java/

Programminghistorian. (2019). *R con MySQL*. Recuperado el 3 de Junio de 2019, de https://programminghistorian.org/en/lessons/getting-started-with-mysql-using-r

Proyectosbeta. (2016). *Logo RStudio*. Recuperado el 27 de Mayo de 2019, de https://proyectosbeta.net/2016/10/rstudio-en-debian-jessie-de-64-bits/

Researchgate. (2019). *Proceso Unificado*. Recuperado el 20 de Mayo de 2019, de https://www.researchgate.net/figure/Flujos-de-trabajo-del-proceso-unificado_fig1_279751761

RSS.nom. (2019). *RSS*. Recuperado el 20 de Mayo de 2019, de https://www.rss.nom.es/

Scribd. (2019). *Astah UML*. Recuperado el 13 de Mayo de 2019, de https://es.scribd.com/document/239437742/Tutorial-ASTAH
Stackoverflow. (2019). *Override*. Recuperado el 3 de Junio de 2019, de https://es.stackoverflow.com/questions/156432/para-que-sirve-la-l%C3%ADnea-override-en-java
Tutorials.jenkov. (2019). *XMLEventReader*. Recuperado el 3 de Junio de 2019, de http://tutorials.jenkov.com/java-xml/stax-xmleventreader.html
Tutorials.jenkov. (2019). *XMLInputFactory*. Recuperado el 3 de Junio de 2019, de http://tutorials.jenkov.com/java-xml/stax-xmlinputfactory.html
Vogella. (2019). *Java XML Tutorial*. Recuperado el 3 de Junio de 2019, de https://www.vogella.com/tutorials/JavaXML/article.html
Wikiepdia. (2019). *MVC*. Recuperado el 27 de Mayo de 2019, de https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador
Wikipedia. (2019). *Arquitectura de Software*. Recuperado el 27 de Mayo de 2019, de https://es.wikipedia.org/wiki/Arquitectura_de_software
Wikipedia. (2019). *Distancia de Levenshtein*. Recuperado el 10 de Junio de 2019, de https://es.wikipedia.org/wiki/Distancia_de_Levenshtein
Wikipedia. (2019). *Git*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/Git
Wikipedia. (2019). *Java*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)
Wikipedia. (2019). *Java Swing*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/Swing_(biblioteca_gr%C3%A1fica)
Wikipedia. (2019). *Lenguaje R*. Recuperado el 13 de Mayo de 2019, de 2019: https://es.wikipedia.org/wiki/R_(lenguaje_de_programaci%C3%B3n)
Wikipedia. (2019). *Logo Git*. Recuperado el 27 de Mayo de 2019, de https://es.m.wikipedia.org/wiki/Archivo:Git-logo.svg
Wikipedia. (2019). *Logo Java*. Recuperado el 27 de Mayo de 2019, de https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)
Wikipedia. (2019). *Logo Microsoft Project*. Recuperado el 27 de Mayo de 2019, de https://es.wikipedia.org/wiki/Microsoft_Project
WIkipedia. (2019). *Microsoft Project*. Recuperado el 20 de Mayo de 2019, de https://es.wikipedia.org/wiki/Microsoft_Project
Wikipedia. (2019). *MySQL*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/MySQL
Wikipedia. (2019). *MySQL Workbench*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/MySQL_Workbench
Wikipedia. (2019). *NetBeans*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/NetBeans
Wikipedia. (2019). *Proceso Unificado*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/Proceso_unificado
Wikipedia. (2019). *Redifusión Web*. Recuperado el 20 de Mayo de 2019, de https://es.wikipedia.org/wiki/Redifusi%C3%B3n_web
Wikipedia. (2019). *RSS*. Recuperado el 6 de Mayo de 2019, de https://es.wikipedia.org/wiki/RSS
Wikipedia. (2019). *RStudio*. Recuperado el 13 de Mayo de 2019, de https://es.wikipedia.org/wiki/RStudio
Wikipedia. (2019). *XML*. Recuperado el 20 de Mayo de 2019, de https://es.wikipedia.org/wiki/Extensible_Markup_Language

APÉNDICES

A. Planificación real detallada

Este proyecto se puede dividir en los siguientes bloques:

1) Análisis.
2) Elección de las herramientas.
3) Extracción de titulares.
4) Limpieza.
5) Almacenamiento.
6) Visualización Estadísticas.

La planificación, estará basada en el desarrollo de estos bloques uno por uno, en orden.
Con la consecución de cada bloque, la aplicación debe de ser capaz de cumplir con unas funcionalidades mínimas que nos permitan avanzar al siguiente bloque.

Lo primero de todo es construir una herramienta que sea un lector del archivo XML que nos proporciona el RSS del periódico digital online. Una vez que la herramienta funcione, se implementará la limpieza de los titulares, es decir, quedarse sólo con las palabras relevantes del titular. Tras esto se procederá al almacenamiento de dichos titulares en una base de datos. Por último, una vez que estén los titulares, guardados, sólo tenemos que utilizarlos para realizar unos gráficos estadísticos.

A continuación, se describen cada una de las tareas realizadas a lo largo de este TFG.
Cada tarea contiene una duración estimada por horas para su realización y el valor real para su desarrollo.

La fecha que se indica la semana en la que se ha realizado.

Actividades realizadas:

Tabla 46. Tarea 1

Tarea 1	**Análisis inicial.**
Fecha	25/02/19 – 01/03/19
Duración Estimada	4 horas.
Duración Real	4 horas.
Predecesoras	0
Detalle	Fase inicial con la elección de la metodología de planificación, así como la gestión de riesgos, gestión de recursos, definición de actividades y calendarización.

Tabla 47. Tarea 2

Tarea 2	**Definición de requisitos.**
Fecha	25/02/19 – 01/03/19
Duración Estimada	4 horas.
Duración Real	4 horas.
Predecesoras	1
Detalle	Obtención de los requisitos que tendrá el proyecto, tanto funcionales, no funcionales y de información; a partir de los datos que se tienen del proyecto.

Tabla 48. Tarea 3

Tarea 3	**Definición de casos de uso**
Fecha	25/02/19 – 01/03/19
Duración Estimada	4 horas.
Duración Real	4 horas.
Predecesoras	2
Detalle	A partir de la información que se tiene del proyecto y de los requisitos ya definidos, creación de los casos de uso, así como los primeros diagramas.

Tabla 49. Tarea 4

Tarea 4	**Definición de modelo de dominio**
Fecha	25/02/19 – 01/03/19
Duración Estimada	4 horas.
Duración Real	4 horas.
Predecesoras	3
Detalle	Elaboración del modelo de dominio, en lenguaje UML, con los datos que se tienen acerca del proyecto.

Tabla 50. Tarea 5

Tarea 5	**Aprender funcionamiento XML**
Fecha	04/03/2019 - 08/03/2019
Duración Estimada	3 horas.
Duración Real	3 horas.
Predecesoras	4
Detalle	Entender qué es un fichero XML. Sus partes, estructura y en definitivas las posibilidades que tiene este tipo de archivo. Aprendizaje a través de un tutorial disponible en Internet (Vogella, 2019). Ayudó a tener el conocimiento necesario para empezar a trabajar con este tipo de formato. El código encontrado en Internet, tomado como base para este proyecto, realizaba la lectura de un archivo XML, por lo que era necesario para poder entender en qué consiste ese código y que funciones realiza tener unos conocimientos básicos acerca de XML.

Tabla 51. Tarea 6

Tarea 6	**Valoración de lenguaje y herramientas a utilizar**
Fecha	04/03/2019 - 08/03/2019
Duración Estimada	10 horas.
Duración Real	15 horas.
Predecesoras	5
Detalle	Tomar la decisión del lenguaje de programación con el que se va desarrollar el programa, así como las distintas herramientas a utilizar a lo largo de todo el proyecto. Se valoraron las siguientes opciones: - Utilización como lenguaje de programación Java. Lenguaje utilizado en la mayoría de prácticas y trabajos a lo largo del grado. Buen dominio. Ventaja en cuanto a

	tiempo, no sería necesaria un excesivo aprendizaje, salvo algunas cosas puntuales. - Utilización de otro lenguaje de programación como puede ser C, Python… Son lenguajes de programación, menos utilizados a lo largo del grado. Podrían suponer una gran pérdida de tiempo en aprendizaje para la realización de ciertas funciones. - Utilizar Visual Studio. Herramienta, no utilizada nunca. Aprendizaje desde cero. Podría llegar a suponer pérdida de tiempo. - Utilizar IDE NetBeans. Herramienta con gran conocimiento, bastante utilizada a lo largo del grado. A mayores, herramienta ya utilizada para la creación de interfaces gráficas - Utilizar IDE Eclipse Herramienta con gran conocimiento, bastante utilizada a lo largo del grado. Sin experiencia de uso con interfaces gráficas. Además de la decisión de con qué herramienta se tienen mayores y mejores conocimientos, se valoró también, a la hora de tomar la decisión, cuál de ellas se podría acoplar mejor a lo que el proyecto en sí iba a necesitar, es decir, qué herramienta permite la extracción de titulares de prensa online. (Palkotools, 2011) y (Dennysjymbo, 2019) Varias páginas en internet mostraban distintas opciones de cómo realizar la lectura de un RSS (Java RSS Feed, 2019), tanto en lenguaje Java cómo C. Tras realizar una valoración de todas estas opciones, la decisión final fue realizar el trabajo, o al menos la parte que conlleva la extracción y limpieza de los titulares de prensa, con el lenguaje Java, ayudado del entorno de programación NetBeans.

Tabla 52. Tarea 7

Tarea 7	**Lectura y extracción de los titulares de una URL**
Fecha	04/03/2019 - 08/03/2019
Duración Estimada	1 hora.
Duración Real	1 hora.
Predecesoras	6
Detalle	Utilización como base del programa el código encontrado en Internet de lectura de un archivo XML. Comprobación de que el código funcionaba y realizaba las funciones esperadas, proporcionando la URL de un archivo RSS de un periódico digital. El resultado fue el esperado, mostrando el resultado en pantalla, imprimiendo cada uno de los titulares que se encontraban el archivo XML del RSS proporcionado, en el momento de la ejecución.

Tabla 53. Tarea 8

Tarea 8	**Lectura y extracción de los titulares de varias URL**
Fecha	11/03/2019 - 15/03/2019
Duración Estimada	1 hora.
Duración Real	1 hora.
Predecesoras	7
Detalle	Tarea pequeña consistente en la modificación del código para que facilitando varias URLs, cada una de un periódico digital diferente, se impriman todos los titulares que se encuentren en el momento de ejecución en cada uno de los archivos RSS proporcionados.

Tabla 54. Tarea 9

Tarea 9	**Ejecución repetida cada cierto tiempo**
Fecha	11/03/2019 - 15/03/2019
Duración Estimada	3 horas.
Duración Real	5 horas.
Predecesoras	8
Detalle	Valoración de las distintas posibilidades (Chuidiang, 2019) y (Poesiabinaria, 2019) de ejecutar el código de una manera periódica, cada un cierto tiempo establecido. Decisión por utilizar la clase Timer de Java, con la que básicamente se ejecuta una porción de código determinada, cada un cierto tiempo especificado en milisegundos. De esta forma, periódicamente, por ejemplo, cada 30 minutos, el programa imprime en pantalla los titulares de las fuentes especificadas.

Tabla 55. Tarea 10

Tarea 10	**Necesidad de almacenar**
Fecha	11/03/2019 - 15/03/2019
Duración Estimada	15 horas.
Duración Real	15 horas.
Predecesoras	9
Detalle	Es necesario conseguir que, de una ejecución del código a la siguiente, sólo, se impriman los nuevos titulares. Es decir, los que han sido publicados, en el periodo de tiempo entre ejecuciones del programa. De esta forma, evitar que el programa imprima todos los titulares que se encuentran en el fichero cada vez, y que sólo sean imprimidos los nuevos. Al existir una limitación, en cuanto al número de titulares en el fichero XML de cada RSS, cada vez que se publica un titular, se añade al fichero, pero se elimina otro, manteniendo constante el número de titulares publicados en el archivo XML. Destacar que el número de titulares es variable para cada uno de los periódicos digitales.

	Para poder llevar a cabo, lo comentado en esta tarea, es necesario un sistema de almacenamiento con el que poder almacenar los titulares. De esta manera, poder consultarlos y si hay algún titular nuevo que no está almacenado, añadirlo. Así se irá incrementando el número de titulares almacenados y cada vez que se publique uno nuevo, no se pierdan los titulares, como ocurría anteriormente.

Tabla 56. Tarea 11

Tarea 11	**Valorar almacenamiento de los titulares**
Fecha	11/03/2019 - 15/03/2019
Duración Estimada	1 hora.
Duración Real	1 hora.
Predecesoras	10
Detalle	Las opciones valoradas, fueron dos. La primera de ellas fue un archivo de texto, en el que ir escribiendo los titulares uno a uno. La segunda es la creación de una base de datos. Debido a las grandes posibilidades que ofrece una base de datos, en cuanto a almacenamiento y consultas. A mayores, pensando en el objetivo final del proyecto, el tener que posteriormente acceder a los titulares almacenados en función de algunos parámetros o valores, lo más lógico era almacenar los titulares en una base de datos. Por tanto, la decisión fue decantarse por la segunda opción. Gracias a la base de datos, se abre un gran abanico de posibilidades, sobre todo a la hora de realizar una representación gráfica de los resultados estadísticos obtenidos.

Tabla 57. Tarea 12

Tarea 12	**Almacenamiento de los titulares**
Fecha	18/03/2019 - 22/03/2019
Duración Estimada	10 horas.
Duración Real	25 horas.
Predecesoras	11
Detalle	Creación de un modelo de dominio en el que basarse para la elaboración de la base de datos. Creación de script SQL para la creación de las tablas necesarias de la base de datos. Valoración de opción de crear una base de datos remota con el servidor de la escuela o una base de datos local. Viendo los pros y contras de cada una de ellas, no obtenía ningún beneficio con la base de datos remota. Finalmente, creación de una base de datos local con MySQL. Conexión de la base de datos con NetBeans. Creación de métodos de inserción de filas a las tablas, necesarios para poder añadir los titulares leídos del RSS correspondiente a su tabla en la base de datos.

Tabla 58. Tarea 13

Tarea 13	**No guardar repetidos**
Fecha	25/03/2019 - 29/03/2019
Duración Estimada	10 horas.
Duración Real	25 horas.
Predecesoras	12
Detalle	Tras solventar el problema del almacenamiento de los titulares con la base de datos, hay que solucionar un nuevo impedimento. Cada vez que se ejecuta el código se van a almacenar los titulares en la base de datos. De ejecución a ejecución es muy frecuente que se lea un gran número de titulares que ya habían sido leídos alguna vez, por tanto, al intentar almacenarles en la base de datos, se va a producir un error de violación de clave primaria, puesto que se va a intentar añadir una instancia, que ya está almacenada. Esta tarea consiste, en idear un procedimiento a partir del cuál, ejecución tras ejecución, sólo se intente añadir a la base de datos aquellos titulares que no estén almacenados, es decir, los titulares que han sido publicados desde la última ejecución. Para ello, para cada titular leído se realiza una consulta a la base de datos, para ver si está almacenado dicho titular o no. Si no está, se almacena y si ya está se pasa al siguiente. A mayores, antes de almacenar el titular en la base de datos hay que comprobar si se ha llegado ya al número máximo de titulares especificado por el usuario. Si ya se ha llegado a ese límite, no se almacenará dicho titular.

Tabla 59. Tarea 14

Tarea 14	**Almacenamiento con más columnas**
Fecha	15/04/2019 - 19/04/2019
Duración Estimada	6 horas.
Duración Real	6 horas.
Predecesoras	13
Detalle	Hasta este punto de TFG sólo se guardaba en la base de datos, para cada titular el título, el periódico dónde ha sido publicado y el campo contenido dentro de la etiqueta de <pubDate>. Dentro de este campo, se encontraban algunos valores, que podemos considerar relevantes y que se pueden guardar por separado como son: el día de la semana, el día del mes, el mes, y la hora de publicación exacta del titular. A mayores, también se añadieron algunos campos que median estadísticas básicas de los titulares de prensa leídos como son el número de caracteres y el número de palabras, para cada uno de los titulares almacenados en la base de datos.

Tabla 60. Tarea 15

Tarea 15	**Limpieza del titular**
Fecha	15/04/2019 - 19/04/2019
Duración Estimada	10 horas.
Duración Real	10 horas.
Predecesoras	14
Detalle	Consiste en la eliminación de aquellas palabras que se pueden considerar menos inútiles o menos relevantes para realizar el análisis estadístico. Entre estas palabras a eliminar, se encuentran las preposiciones, conjunciones, disyunciones así cómo los signos ortográficos. De esta forma, en una nueva columna en la base de datos se añadiría el titular que se puede considerar como limpio.

Tabla 61. Tarea 16

Tarea 16	**Elaboración Consultas SQL**
Fecha	22/04/2019 - 26/04/2019
Duración Estimada	10 horas.
Duración Real	10 horas.
Predecesoras	15
Detalle	Valoración de los gráficos a realizar en un futuro y con ello la elaboración de las consultas SQL, cuyo resultado servirán a la herramienta gráfica para realizar las representaciones correspondientes.

Tabla 62. Tarea 17

Tarea 17	**Creación de interfaz**
Fecha	22/04/2019 - 03/05/2019
Duración Estimada	25 horas.
Duración Real	40 horas.
Predecesoras	16
Detalle	La interfaz gráfica será implementada siguiendo el patrón Modelo-Vista-Controlador. Consistirá de tres vistas: 1) El usuario podrá seleccionar el periódico y el tipo de información que se extraerá, así como el número máximo de titulares que se almacenarán en la base de datos. 2) El usuario podrá ver el porcentaje del experimento completado, así como los últimos titulares que han sido añadidos a la base de datos. 3) Muestra de algunos gráficos estadísticos sencillos.

Tabla 63. Tarea 18

Tarea 18	**Implementación búsqueda de palabras**
Fecha	06/05/2019 - 10/05/2019
Duración Estimada	10 horas.
Duración Real	10 horas.
Predecesoras	17
Detalle	Añadir la posibilidad de que el usuario pueda elegir una o varias palabras que el programa tendrá en cuenta a la hora de seleccionar los titulares. La idea es que el usuario, cuando acabe el experimento pueda obtener información detallada de los titulares que contengan esa palabra o palabras especificadas.

Tabla 64. Tarea 19

Tarea 19	**Implementación de palabras negativas**
Fecha	06/05/2019 - 10/05/2019
Duración Estimada	5 horas.
Duración Real	5 horas.
Predecesoras	18
Detalle	Elaboración de una lista con palabras que se pueden considerar negativas como pueden ser “violencia” o “asesinato”. El fin, es hacer una distinción de aquellos titulares que contengan alguna de estas palabras consideradas negativas y hacer una comparación de la cantidad de estos titulares por fuente o tipo de información. No se trata de una lista de palabras cerrada. Se implementa un método para comprobar *palabras similares* a las dadas, mediante una comparación entre cadenas de caracteres con la distancia de Levenshtein.

Tabla 65. Tarea 20

Tarea 20	**Implementación de palabras positivas**
Fecha	06/05/2019 - 10/05/2019
Duración Estimada	5 horas.
Duración Real	5 horas.
Predecesoras	19
Detalle	De forma análoga a la implementación de palabras negativas.

Tabla 66. Tarea 21

Tarea 21	**Realización de gráficos con NetBeans**
Fecha	13/05/2019 - 17/05/2019
Duración Estimada	10 horas.
Duración Real	15 horas.
Predecesoras	20
Detalle	Elaboración de gráficos que serán mostrados una vez finalizado el experimento, en la tercera vista. Entre estos gráficos se encuentran: 1) Media de palabras de los titulares por fuente y tipo de información. 2) Media de caracteres de los titulares por fuente y tipo de información. 3)Número de titulares almacenados por fuente y tipo de información.

Tabla 67. Tarea 22

Tarea 22	**Elección de programa para realizar gráficos**
Fecha	20/05/2019 - 24/05/2019
Duración Estimada	5 horas.
Duración Real	5 horas.
Predecesoras	20
Detalle	La idea era encontrar un programa que permitiera la representación de gráficos para demostrar la potencia de la herramienta conectando los resultados obtenidos en la base de datos. La primera idea fue utilizar el lenguaje R, que es un lenguaje estadístico con gran capacidad de representación y visualización de gráficos. R tiene el inconveniente de que no permite realizar gráficos interactivos, por lo que a mayores se decide buscar un programa que permita esta opción. Se valoran programas como QLik, Tableau o Power BI. Estos 3 programas cumplirían perfectamente las expectativas. La elección final fue utilizar Power BI, debido a que su versión gratuita no era por un plazo de tiempo, sino que estaba con funcionalidad limitada. Sin embargo, con las opciones gráficas que muestra la versión gratuita, es más que suficiente para demostrar la potencia de la herramienta, con una gran disponibilidad de gráficos y interactivos.

Tabla 68. Tarea 23

Tarea 23	**Realización de gráficos con R**
Fecha	20/5/2019 - 24/05/2019
Duración Estimada	4 horas.
Duración Real	4 horas.
Predecesoras	22

Detalle	Lo primero de todo, realizar la conexión de la base de datos con el programa RStudio, mediante el uso de los paquetes disponibles necesarios. (Programminghistorian, 2019) Una vez, disponibles los datos dentro del programa, realización de gráficos apropiados para este tipo de datos, entre los que se encuentra el WordCloud.

Tabla 69. Tarea 24

Tarea 24	**Lectura de archivos CSV**
Fecha	20/05/2019 - 24/05/2019
Duración Estimada	20 horas.
Duración Real	30 horas.
Predecesoras	23
Detalle	Con la finalidad de que el usuario de la aplicación pueda modificar el listado de palabras negativas, positivas, así como aquellas palabras con las que desea realizar una búsqueda. Esta modificación consistiría en la eliminación de alguna palabra, así como la inserción de alguna nueva. Con esta implementación, mediante el uso de 3 archivos CSV (positivas, negativas, búsqueda), el usuario podrá modificar en ellos las palabras a su elección y mediante la interfaz de usuario podrá cargarlos, siendo añadidas las palabras a sus correspondientes tablas a la base de datos para su fututo procesamiento.

Tabla 70. Tarea 25

Tarea 25	**Implementar procesamiento de palabras**
Fecha	27/05/2019 - 31/05/2019
Duración Estimada	10 horas.
Duración Real	15 horas.
Predecesoras	24
Detalle	Funcionalidad de procesar las palabras, es decir, en función de la información que el usuario haya proporcionado con los archivos CSV con las palabras para buscar, positivas y negativas. Se revisarán las palabras almacenadas en la base de datos, con las palabras proporcionadas y se establecerán si son de búsqueda, positivas o negativas, o ninguna de estas tres, con los correspondientes campos en la tabla de Palabras de la base de datos.

Tabla 71. Tarea 26

Tarea 26	**Aprendizaje de Power BI**
Fecha	03/06/2019 - 07/06/2019
Duración Estimada	4 horas.
Duración Real	4 horas.
Predecesoras	25

Detalle	Debido a que se trata de un programa del que no se tiene conocimiento acerca de su utilización. Uso de tutoriales y ejemplos para obtener una capacidad suficiente para la realización de gráficos.

Tabla 72. Tarea 27

Tarea 27	**Realización de gráficos con Power BI**
Fecha	07/06/2019 - 07/06/2019
Duración Estimada	4 horas.
Duración Real	4 horas.
Predecesoras	26
Detalle	Lo primero de todo, realizar la conexión de la base de datos con el programa Power BI Desktop (Microsoft, 2019) y (Elfuturodelosdatos, 2019). Una vez, disponibles los datos dentro del programa, realización de gráficos apropiados para este tipo de datos, aprovechando las opciones que dispone este programa para la realización de gráficos interactivos y de Dashboards.

Tabla 73. Tarea 28

Tarea 28	**Pruebas y resolución de fallos**
Fecha	Durante todo el desarrollo del proyecto
Duración Estimada	40 horas
Duración Real	50 horas
Predecesoras	-
Detalle	Pruebas de ejecución del programa, durante un tiempo en busca de algún fallo. Tarea que consiste en dejar ejecutándose el programa durante un tiempo prolongado de tiempo, periódicamente. Comprobación de que la base de datos se rellenaba tal cuál había sido ideada. Se realiza siempre, cuando se ha añadido una nueva funcionalidad o modificado alguna anterior.

Tabla 74. Tarea 29

Tarea 29	**Redacción de la memoria del proyecto**
Fecha	06/05/2019 -27-06-19
Duración Estimada	50 horas
Duración Real	80 horas
Predecesoras	-
Detalle	Redacción de la memoria y documentación de este TFG.

Diagrama

A continuación, se muestra el diagrama de Gantt correspondiente a las tareas realizadas a lo largo de este TFG.

Conviene aclarar que al igual, que el diagrama de la planificación inicial, la tarea que se corresponde con las pruebas se ha realizado periódicamente desde el inicio hasta el fin del desarrollo de código, con la modificación o adición de una funcionalidad.

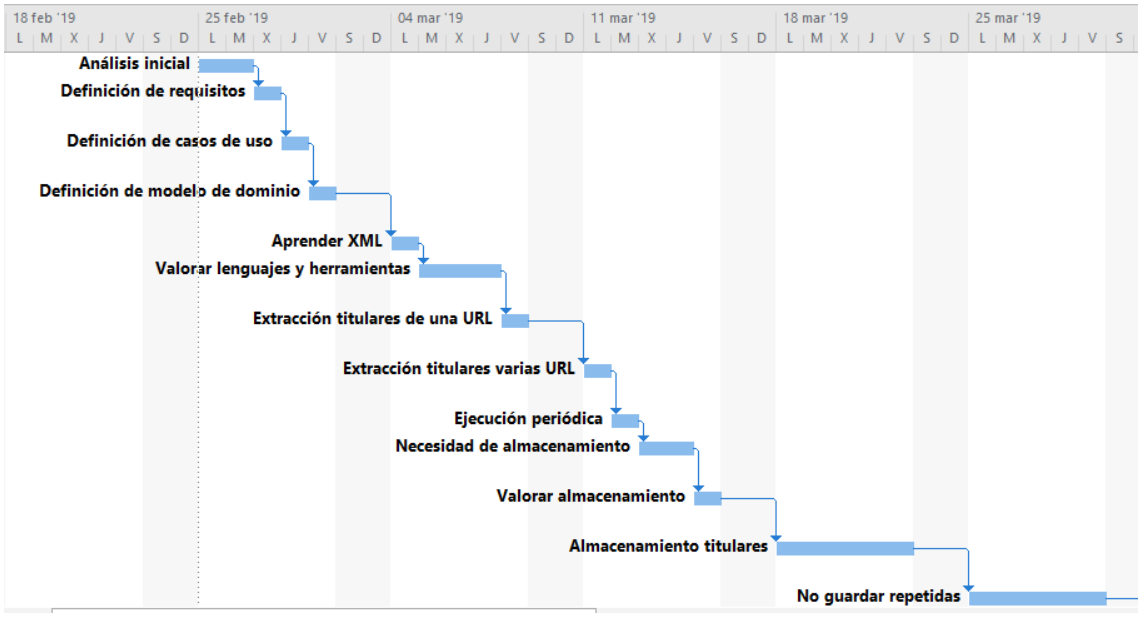


Figura 59. Diagrama Gantt Planificación Real 1

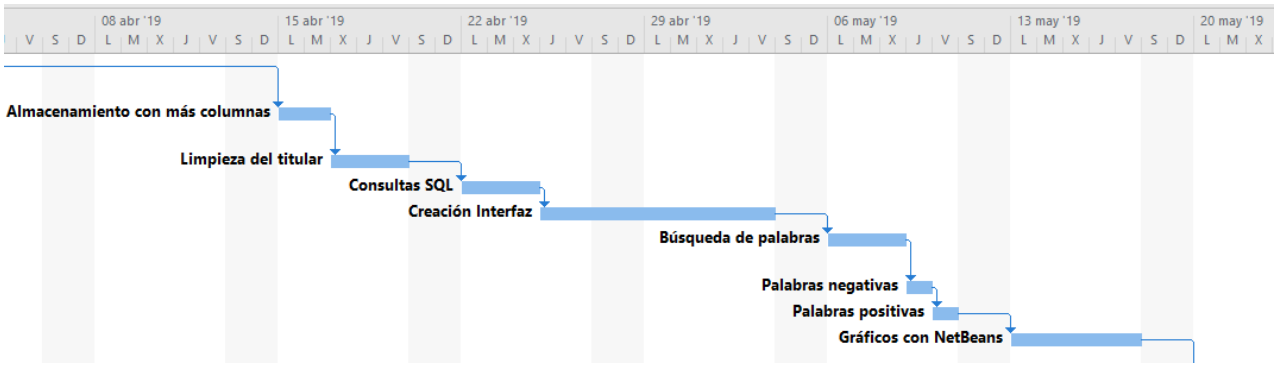


Figura 60. Diagrama Gantt Planificación Real 2

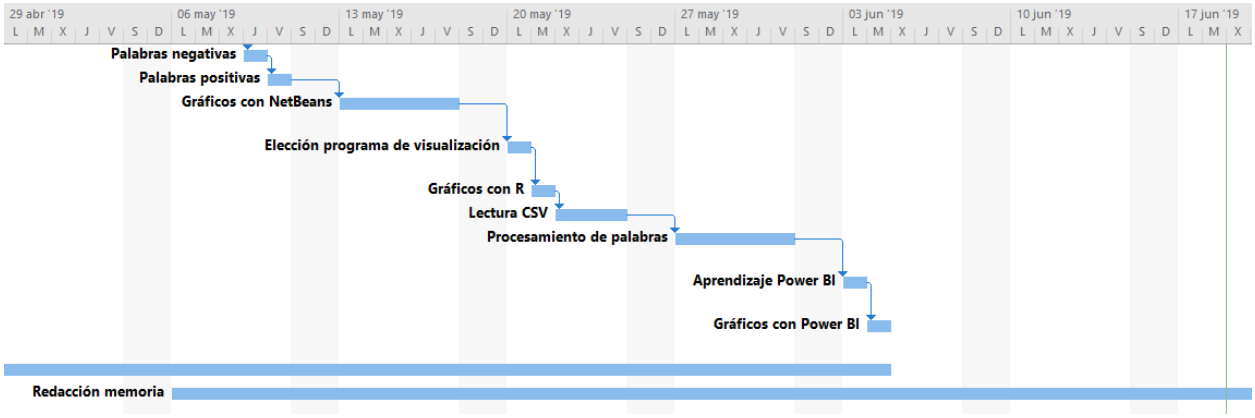


Figura 61. Diagrama Gantt Planificación Real 3

B. Conexión con Power BI

En este apartado se explicará como realizar la conexión de la base de datos con el programa de visualización de gráficos Power BI.

Es necesario tener la base de datos creada en el servidor local, y con algunos datos guardados, es decir, para poder realizar los gráficos, la base de datos no puede estar vacía.

En primer lugar, como es lógico, es descargar el programa de su página web https://powerbi.microsoft.com/es-es/downloads/ eligiendo la versión gratuita.

Figura 62. Descarga de Power BI

Una vez descargado e instalado, sólo hay que abrir un proyecto nuevo.
El usuario deberá elegir en la barra de herramientas la opción de "Obtener datos".

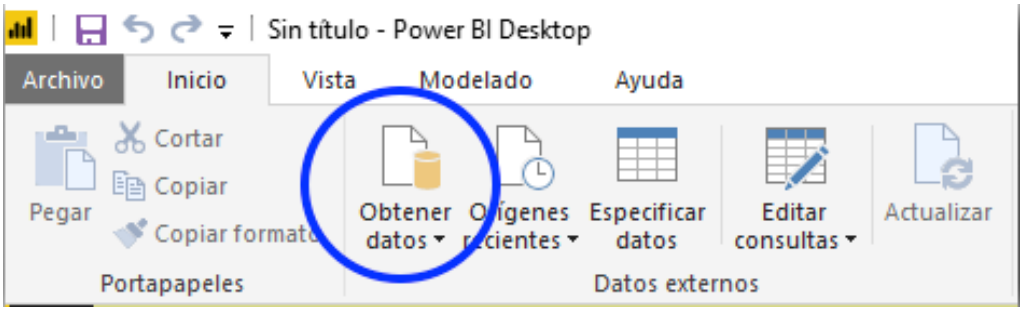


Figura 63. Obtener datos en Power BI

En la ventana que se despliega, hay que elegir la opción de Base de datos MySQL.

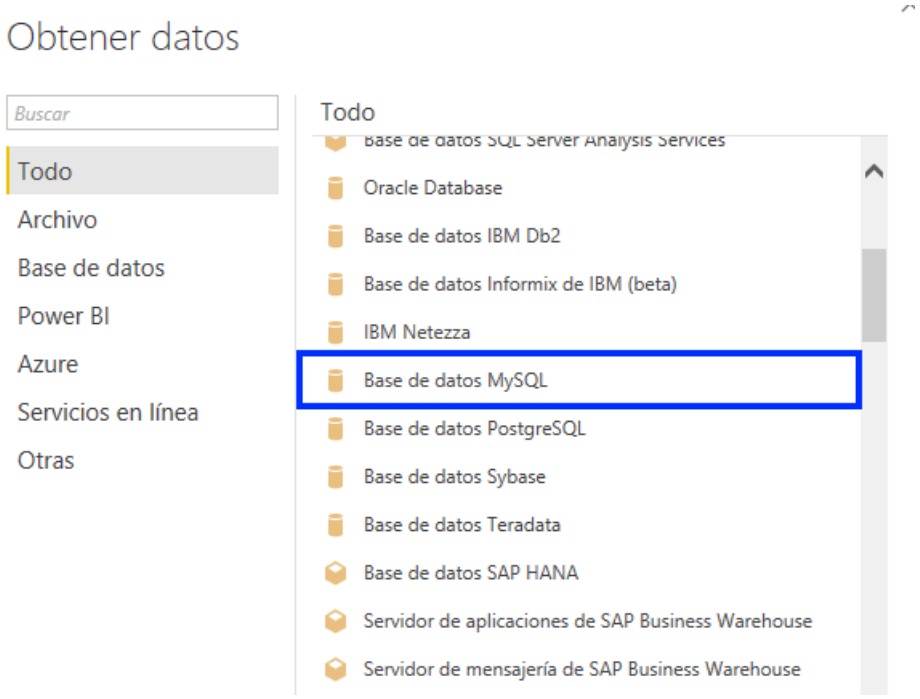


Figura 64. MySQL en Power BI

El programa pedirá que el usuario introduzca el servidor donde se encuentra la base de datos, así como el nombre de la misma. Se debe introducir "localhost" y "tfg" respectivamente.

Figura 65. Seleccionar base de datos

En este punto, se abrirá una nueva ventana en la que se debe seleccionar la pestaña de "Base de datos", donde hay que especificar el nombre de usuario y la contraseña de la base de datos. En este caso, hay que introducir "root" y "bicicleta" respectivamente.

Por último, antes de conectar hay que elegir el nivel donde aplicar la configuración dada. El usuario debe seleccionar la opción "locahost;tfg".

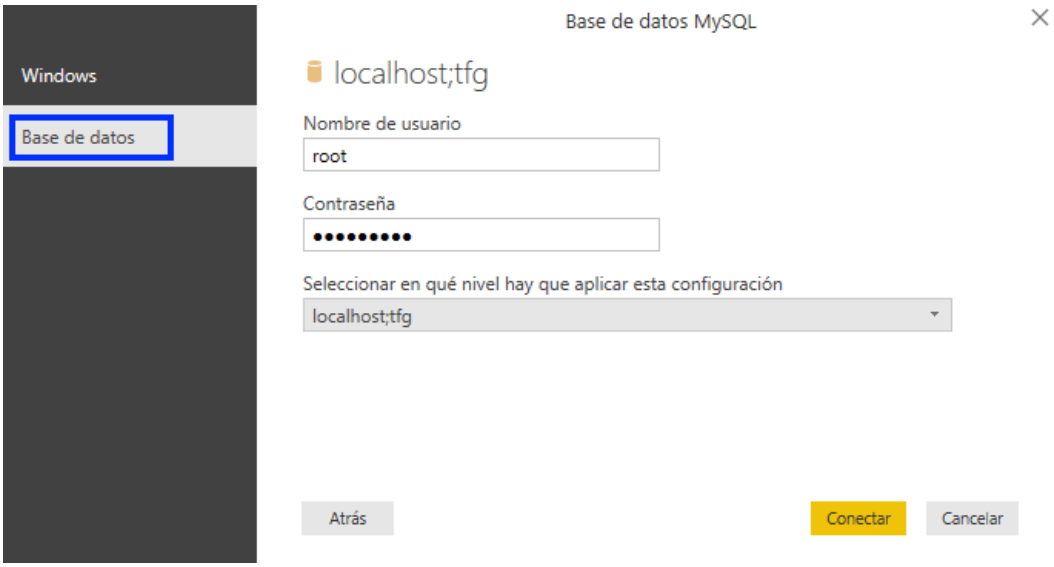


Figura 66. Introducir usuario y contraseña

Tras darle al conectar, se podrá ver una vista como la siguiente en la que ya aparecen las tablas que se encuentran en la base de datos. Este proceso tarda un poco, mientras realiza la conexión con la base de datos.

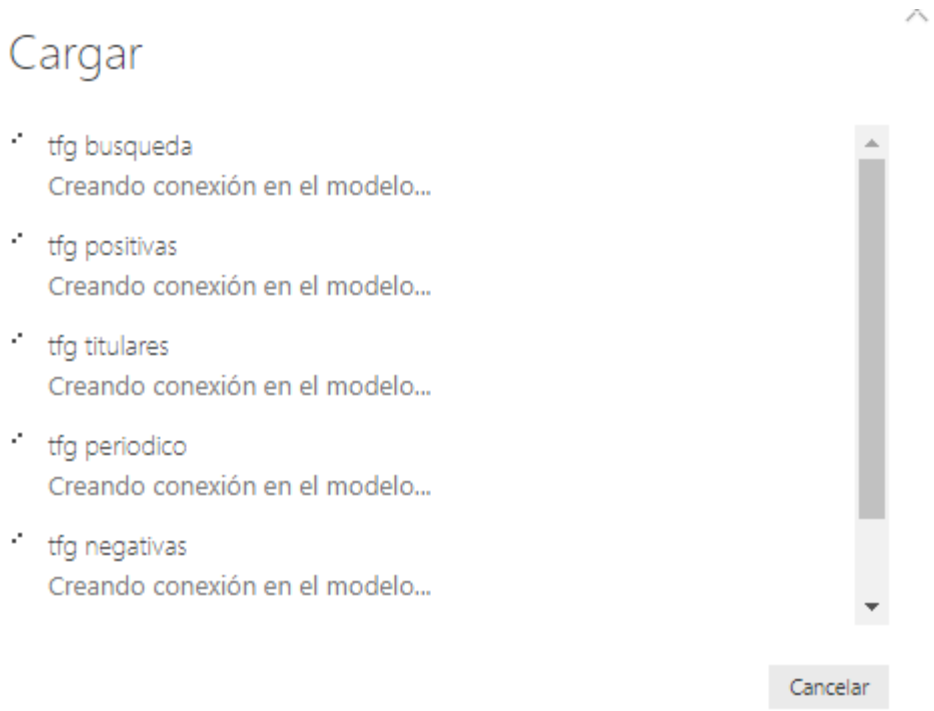


Figura 67. Conexión con la base de datos

Tras finalizar la conexión, se muestra la opción de elegir que tablas nos interesan y cuáles no. Para poder utilizar toda la información almacenada en la base de datos, se deben seleccionar las 6 tablas existentes.

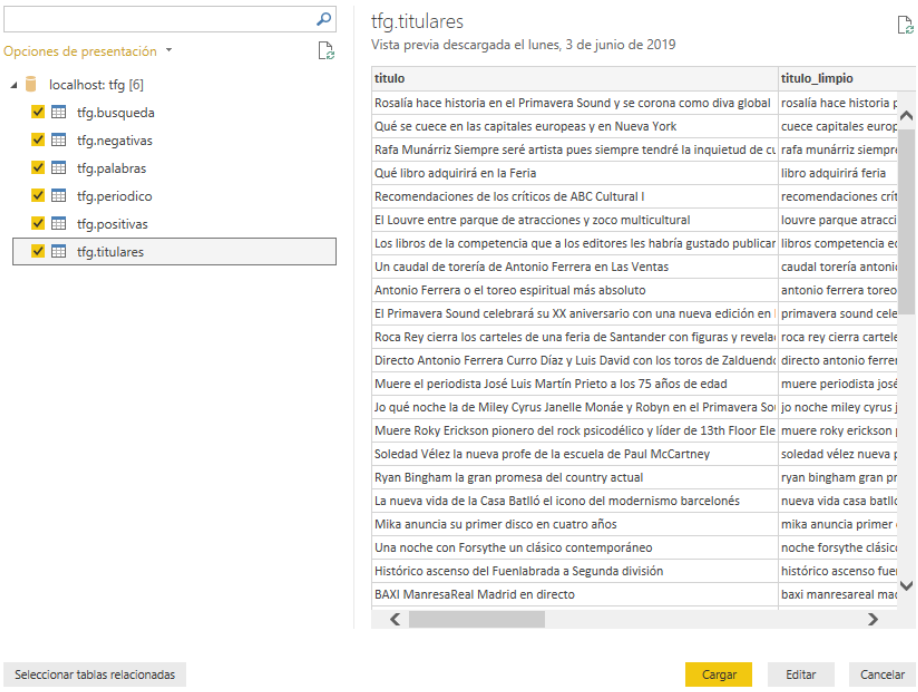


Figura 68. Selección de tablas

Tras realizar el proceso de carga de datos al programa, el usuario podrá ver en la parte derecha de la pantalla, una opción como la siguiente imagen donde aparece una lista con todas las tablas cargadas, así como sus columnas.

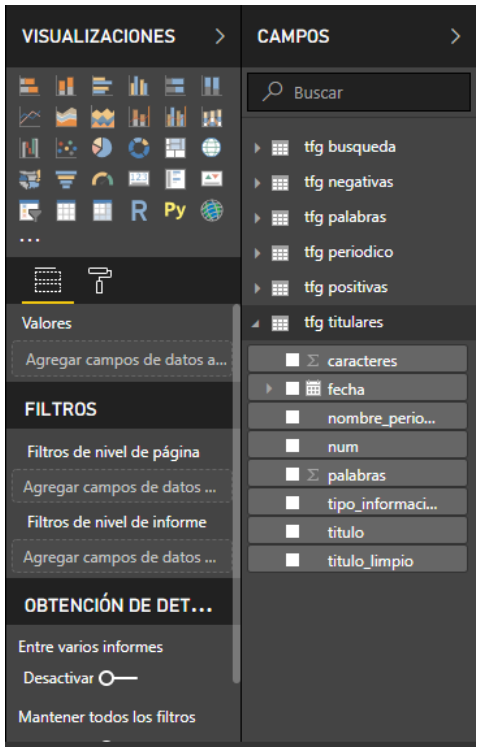


Figura 69. Datos cargados

En este momento, el usuario ya puede seleccionar aquellos datos que le interesen para la realización de gráficos.

Para poder utilizar este programa conviene realizar un tutorial, para aprender sus funciones básicas para la visualización de gráficos.

Conviene tener en cuenta, que a medida que se actualiza la base de datos añadiendo nueva información es necesario actualizar los datos cargados en el programa Power BI.

Por tanto, si se quiere tener la ultima información existente en la base de datos, es necesario actualizar la información. Para ello el usuario simplemente el usuario tendrá que pulsar en el botón de "Actualizar" en la barra de herramientas del programa.

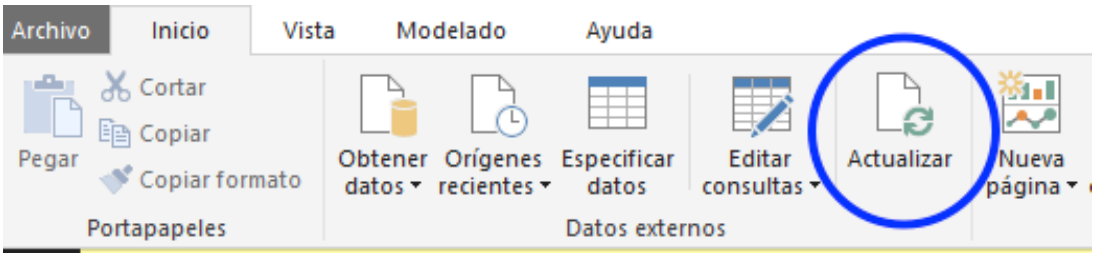


Figura 70. Actualizar datos

Por ultimo, se muestra un ejemplo de visualización de gráficos. La siguiente imagen, muestra un pequeño dashboard interactivo formado por un gráfico de tarta y un gráfico de barras.

El gráfico de tarta, muestra la proporción en cuanto a número de titulares por periódico.

El gráfico de barras muestra el número de titulares, por tipo de información.

El gráfico es interactivo, y se puede ver, por ejemplo, la proporción de numero de titulares de cada uno de los periódicos en el gráfico de barras, agrupado por tipo de información.

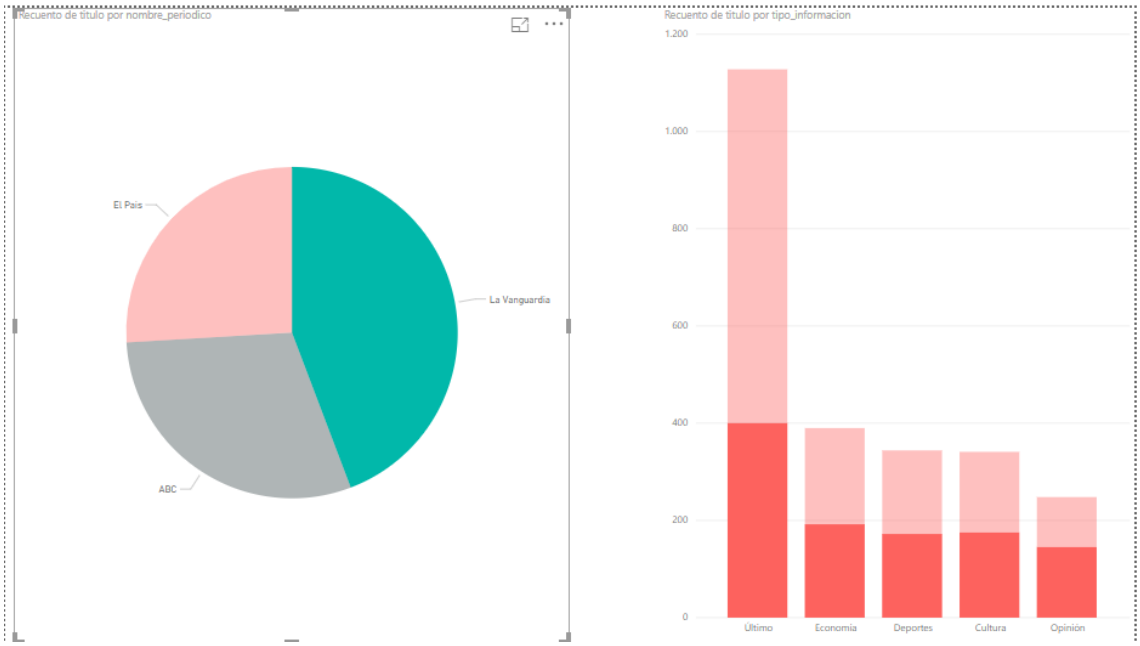


Figura 71. Dashboard

C. Conexión con RStudio

En este apartado se muestra el Script utilizado para utilizar los datos almacenados en la base de datos, desde otro programa de visualización de datos como es RStudio utilizando el lenguaje R.

En este Script se realiza la conexión especificando el usuario, el servidor local por defecto, el nombre de la base de datos y la contraseña.

A continuación, se muestra un ejemplo de como realizar una consulta y guardar los datos que devuelve esa consulta en una variable.

```
#Instalación de librerías necesarias
#install.packages("dbConnect")
#install.packages("wordcloud")
#library(RMySQL)
#library(dbConnect)
#library(wordcloud)

con=dbConnect(MySQL(),user="root",password="bicicleta",dbname="tfg")

dbListTables(con)
encoding <- if(grepl(pattern = 'utf8|utf-8',x = Sys.getlocale(),ignore.case = T)) 'utf8' else 'latin1'
dbGetQuery(con,paste("SET names",encoding))
dbGetQuery(con,paste0("SET SESSION character_set_server=",encoding))
dbGetQuery(con,paste0("SET SESSION character_set_database=",encoding))

consulta3 <- "SELECT palabra, suma FROM (select palabra ,sum(contador) as suma
              from Palabras group by palabra) as myalias where suma>1;"
Contador_Palabras<-dbGetQuery(con,consulta3)
str(Contador_Palabras)
```

Figura 72. Script RStudio

El ejemplo que se muestra es el de un WordCloud, un gráfico muy intuitivo para ver la frecuencia con la que aparecen las palabras.

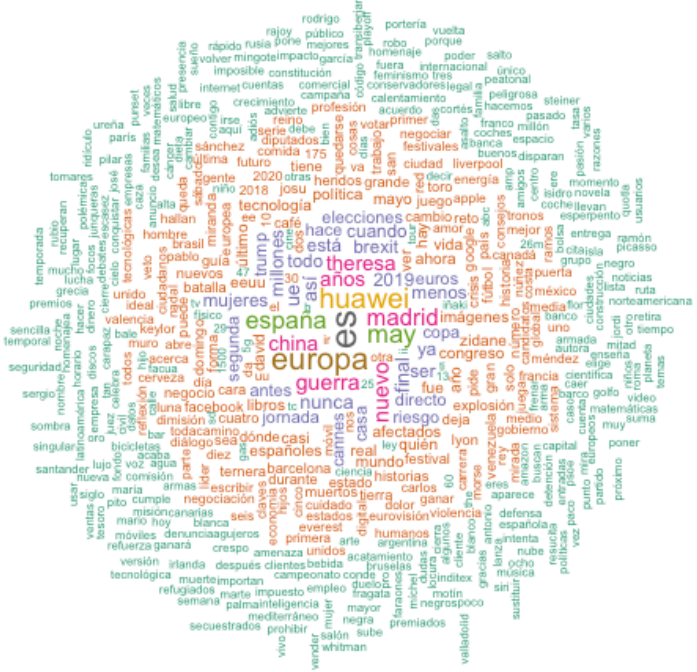


Figura 73. WordCloud

D. Manual de Usuario

Se incluye el manual de instrucciones para el usuario explicando las funcionalidades y opciones que presenta la herramienta desarrollada en este TFG.

Se muestra la información, por apartados que se corresponden con las vistas disponibles en la interfaz de usuario del programa.

Selección de información

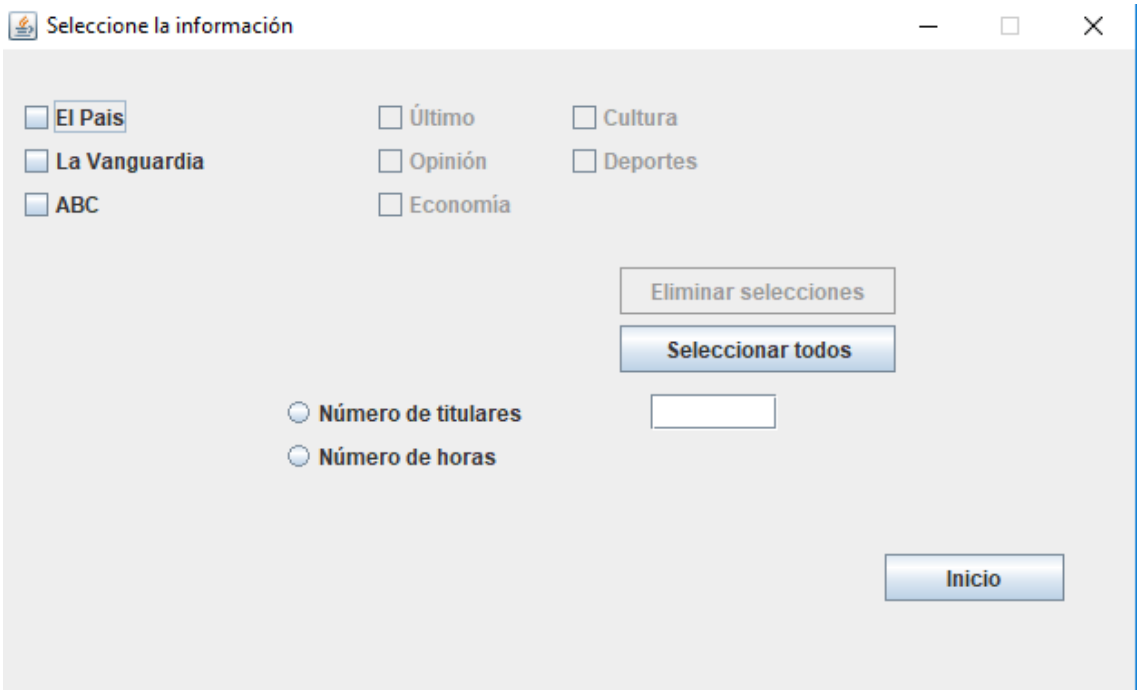


Figura 74. Vista para seleccionar la información

En la parte izquierda se muestran 3 CheckBoxes para poder seleccionar, los periódicos. Es necesario seleccionar al menos uno, para poder comenzar el proceso de extracción de titulares.

En la parte derecha, aparecen otros 5 CheckBoxes para seleccionar el tipo de información. Estos CheckBoxes aparecerán deshabilitados mientras no haya ningún periódico seleccionado. Cuando haya al menos un periódico, se habilitarán. Es necesario seleccionar al menos uno, para poder comenzar con el proceso de extracción de titulares.

Debajo de los selectores del tipo de información se incluyen dos botones. Estos botones se han incluido para facilitar la experiencia de uso del usuario con la interfaz.

El primero de ellos consiste en “Eliminar selecciones”. Cuando el usuario lo pulsa se deselecciona todas aquellas opciones previamente marcadas, tanto de tipo de información como de periódico. Este botón sólo se mostrará habilitado para su uso cuando al menos el usuario tenga alguna selección realizada.

El segundo botón trata de “Seleccionar todos.” Cuando el usuario lo pulse, se seleccionarán todas las opciones posibles, es decir, todos los periódicos y todos los tipos de información. De esta forma se consigue que con un solo click el usuario pueda preparar el programa para la extracción de toda la información posible.

En la parte inferior, en el centro se encuentran 2 RadioButtons, junto con un campo de texto. Esto sirve para que el usuario pueda seleccionar el modo de finalización del proceso de extracción.

Se muestran 2 opciones la primera de ellas es con el “Número de titulares”. Cuando el usuario seleccione esta opción deberá introducir el número de titulares que, a partir de la hora de inicio de ejecución del programa, el usuario quiere que se almacenen.

La segunda opción es más simple, con el "Número de horas", el usuario introduce en el campo de texto el número de horas que quiere que el programa almacene titulares. Cuando se cumpla el tiempo, el proceso de extracción finalizará.

Por último, se encuentra el botón de Inicio, con el que el usuario puede dar comienzo al proceso de extracción de titulares. Es necesario que el usuario haya proporcionado toda la información necesaria, si no es así, el programa mostrará un mensaje indicando qué información falta de introducir.

Resultados en directo

Una vez que el usuario haya proporcionado toda la información necesaria y haya pulsado en el botón de Inicio se mostrará la siguiente vista.

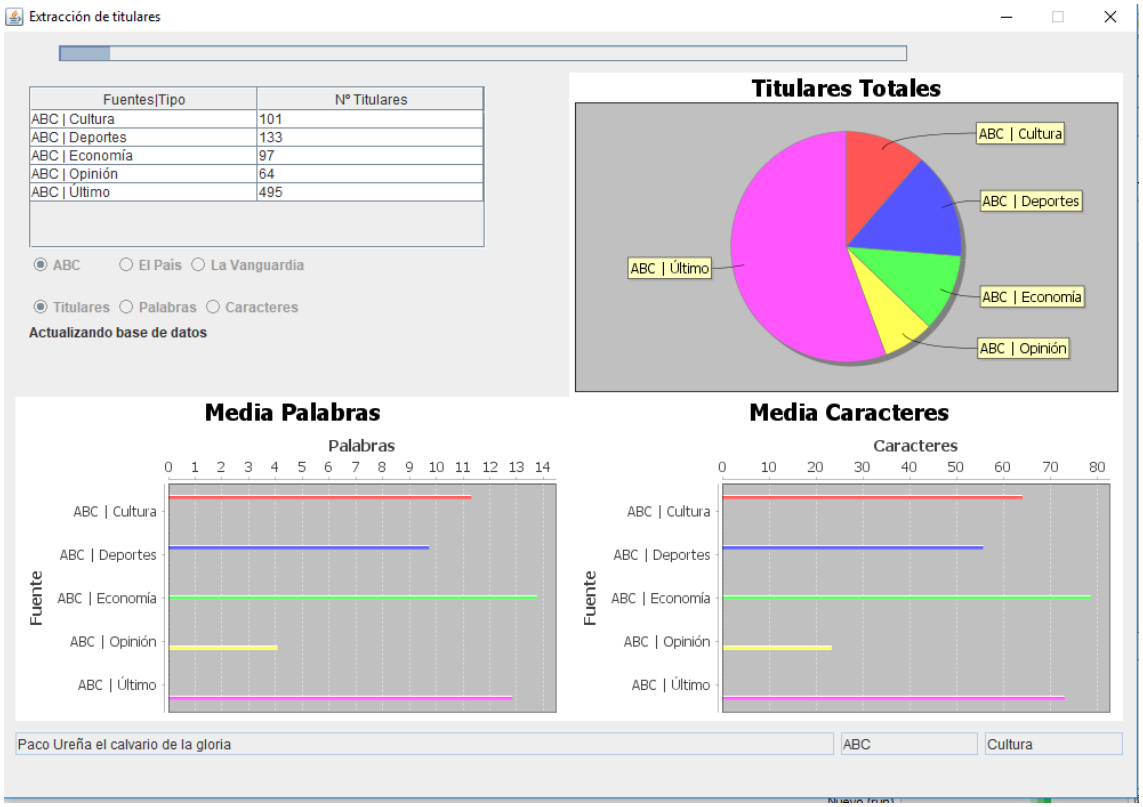


Figura 75. Vista con los resultados en directo

En esta vista se muestra un Dashboard, compuesto por una tabla, un PieChart y 2 gráficos de barras.

La tabla muestra dos selectores formados por tres RadioButtons cada uno.
Con el primero de ellos el usuario puede seleccionar el periódico del que se visualizará la información almacenada en la base de datos. Con este selector se cambia el Dashboard entero.

Con el segundo selector, se puede ver qué información se quiere visualizar en la tabla, para ver el valor numérico exacto que se representará en los gráficos.

Destacar que estos selectores, sólo se podrá interactuar con ellos mientras no se está actualizando la base de datos. La actualización de la base de datos se realiza periódicamente, cada un cierto número de tiempo especificado. Durante unos segundos, el usuario verá en la interfaz un mensaje indicando que se está actualizando la base de datos. Una vez, se termine de actualizar la base de datos, el usuario tendrá la capacidad de modificar la información representada.

En la parte superior se puede observar una barra de progresa con la que el usuario se puede hacer una idea del tiempo de ejecución que resta al programa ya sea por número de horas que han transcurrido o por nuevos titulares que se han almacenado en la base de datos.

En la parte inferior, se muestra el último titular que ha sido añadido a la base de datos, junto con la información del periódico y tipo, dónde ha sido publicado.

Resultados

Cuando finalice el proceso de extracción por cualquiera de los dos métodos límite de ejecución se mostrará la siguiente vista.

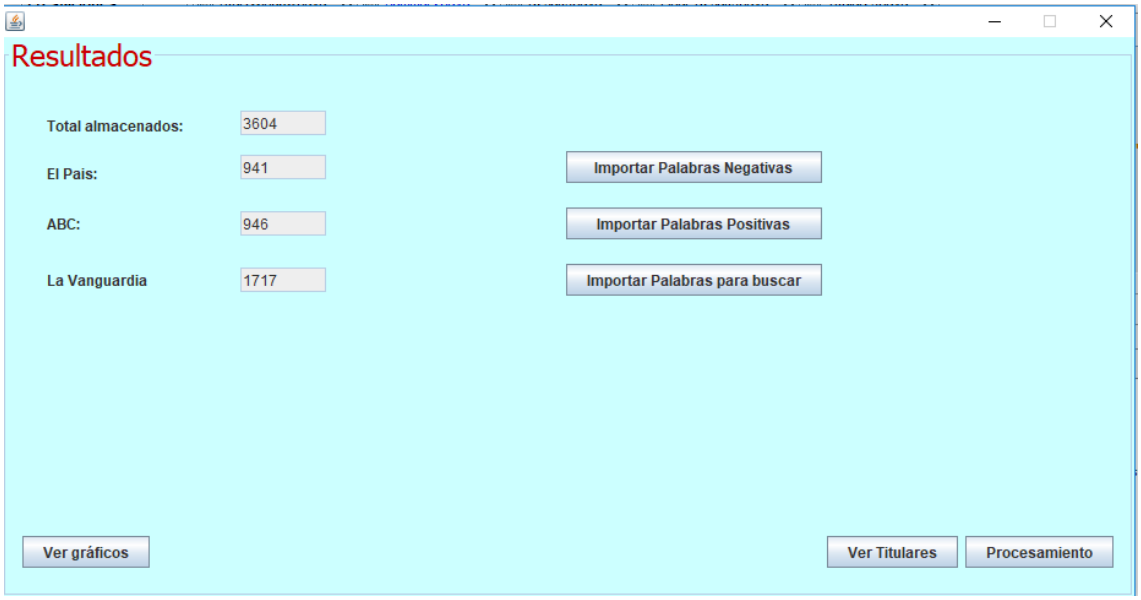


Figura 76. Vista con resumen de resultados

En esta vista se podrá ver un pequeño resumen, del número de titulares que se encuentran almacenados en ese momento en la base de datos y divididos por cada uno de los tres periódicos.

En esta vista el usuario podrá realizar algunas acciones que son las siguientes:

- Ver gráficos

  Al pulsar este botón el usuario podrá ver los gráficos que se mostraban mientras el proceso de extracción, con una vista muy similar.

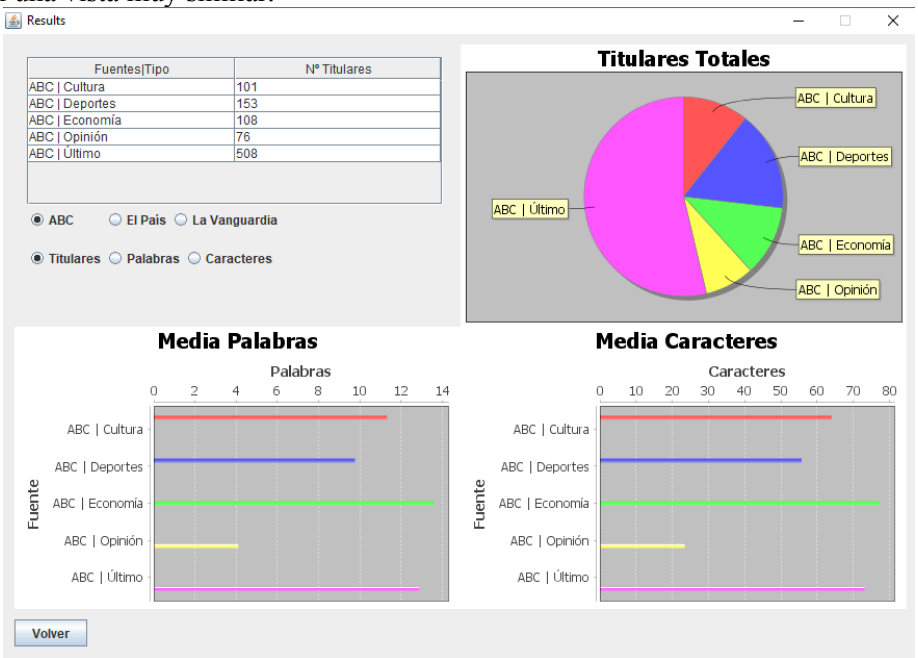


Figura 77. Vista para Ver Gráficos

En este caso, ya no se muestra la barra de progreso ni el último titular que ha sido almacenado en la base de datos. El usuario puede volver a la vista de resultados pulsando en el botón "Volver".

- Importar palabras positivas

  El usuario puede introducir una serie de palabras que han sido consideradas positivas, mediante un archivo CSV guardado en la ruta:

  C:/ProgramData/MySQL/**MySQL Server 8.0**/**Uploads**/positivas.csv

  Cuando se importe el archivo, se mostrará un mensaje de que el archivo ha sido importado.

- Importar palabras negativas

  El usuario puede introducir una serie de palabras que han sido consideradas positivas, mediante un archivo CSV guardado en la ruta:

  C:/ProgramData/MySQL/**MySQL Server 8.0**/**Uploads**/negativas.csv

  Cuando se importe el archivo, se mostrará un mensaje de que el archivo ha sido importado.

- Importar palabras para buscar:

  El usuario puede introducir una serie de palabras que han sido consideradas positivas, mediante un archivo CSV guardado en la ruta:

  C:/ProgramData/MySQL/**MySQL Server 8.0**/**Uploads**/busqueda.csv

  Cuando se importe el archivo, se mostrará un mensaje de que el archivo ha sido importado.

- Procesamiento:

  Una vez que el usuario haya importado los archivos, el usuario podrá realizar el procesamiento de las palabras. Si no se importan los archivos, se utilizará la información que se encuentre en ese momento almacenada en la base de datos. Por tanto, si el usuario quiere modificar alguna palabra positiva, negativa o de búsqueda, debe importar los archivos para actualizar la base de datos.
  Una vez que el procesamiento finalice, se mostrará al usuario un mensaje indicándolo.

- Ver titulares:

  Una vez realizado el procesamiento, el usuario tendrá la capacidad de ver qué titulares contienen aquellas palabras que han sido proporcionadas mediante los archivos CSV.
  Se mostrará una vista como la siguiente, dónde se podrán ver todos aquellos titulares que contienen una palabra de búsqueda o que han sido considerados positivos o negativos.
  Con un selector con RadioButtons, el usuario podrá elegir qué información quiere visualizar.
  Aparecerá en la parte inferior el número de titulares totales junto con el número de titulares positivos, negativos y de búsqueda.
  También puede volver a la vista de Resultados, pulsando en "Volver", para por ejemplo poder hacer un nuevo procesamiento de palabras.

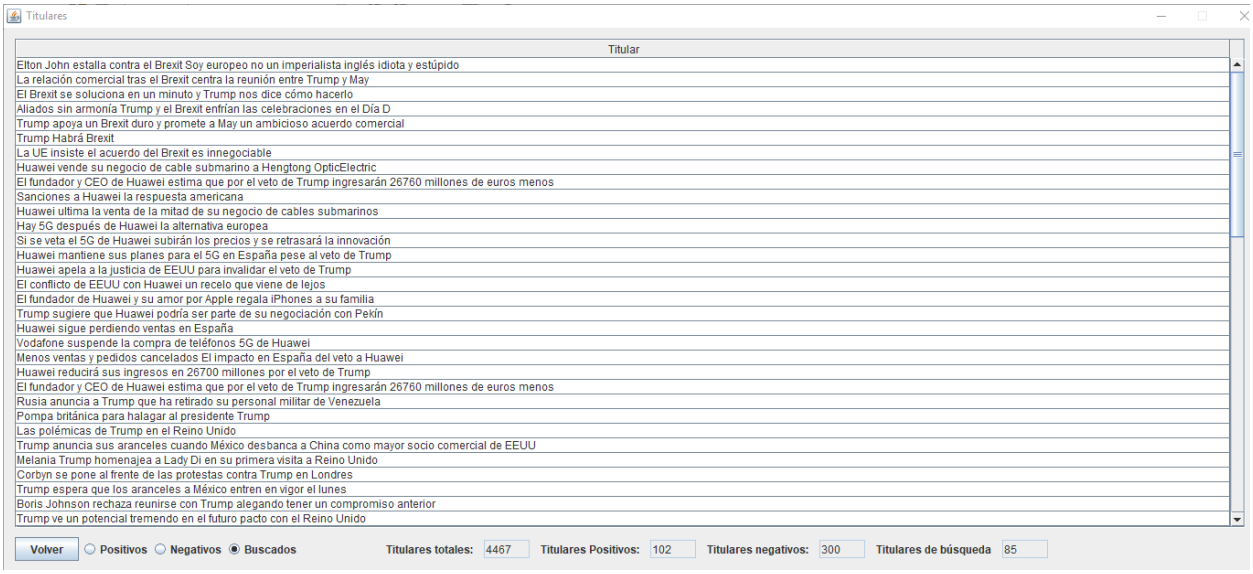


Figura 78. Vista para Ver Titulares

E. Manual de instalación:

Para poder ejecutar el programa, lo primero de todo es configurar la base de datos MySQL.

Para ello teniendo instalado MySQL hay que crear una base de datos con las siguientes características:

- Usuario: root
- Password: bicicleta
- Nombre de la base de datos: tfg

Una vez que esté la base de datos creada, se ejecuta el script SQL que se proporciona.

En ultima instancia, hay que importar el proyecto que se encuentra en el CD a NetBeans, mediante la opción “Import from ZIP” o “Open project”.

Una vez importado hay que cargar las bibliotecas proporcioandas en el CD, con la opción de NetBeans “Import from JAR” en el proyecto.

Finalmente, sólo falta ejecutar la aplicación.

F. Anexos de código

Anexo 1: Clase Java denominada *FeedMessage* utilizada como base para el desarrollo de este TFG. Se hace referencia a este anexo en el capítulo de Implementación.

URL: https://www.vogella.com/tutorials/RSSFeed/article.html

```
package de.vogella.rss.model;

/*
 * Represents one RSS message
 */
public class FeedMessage {

    String title;
    String description;
    String link;
    String author;
    String guid;

    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
    public String getLink() {
        return link;
    }
    public void setLink(String link) {
        this.link = link;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getGuid() {
        return guid;
    }
    public void setGuid(String guid) {
        this.guid = guid;
    }
    @Override
    public String toString() {
        return "FeedMessage [title=" + title + ", description=" + description
                + ", link=" + link + ", author=" + author + ", guid=" + guid
                + "]";
    }
}
```

Anexo 2: Clase Java denominada *Feed* utilizada como base para el desarrollo de este TFG. Se hace referencia a este anexo en el capítulo de Implementación.

URL: https://www.vogella.com/tutorials/RSSFeed/article.html

```
package de.vogella.rss.model;

import java.util.ArrayList;
import java.util.List;

/*
 * Stores an RSS feed
 */
public class Feed {

    final String title;
    final String link;
    final String description;
    final String language;
    final String copyright;
    final String pubDate;

    final List<FeedMessage> entries = new ArrayList<FeedMessage>();

    public Feed(String title, String link, String description, String language,
            String copyright, String pubDate) {
        this.title = title;
        this.link = link;
        this.description = description;
        this.language = language;
        this.copyright = copyright;
        this.pubDate = pubDate;
    }

    public List<FeedMessage> getMessages() {
        return entries;
    }

    public String getTitle() {
        return title;
    }

    public String getLink() {
        return link;
    }

    public String getDescription() {
        return description;
    }

    public String getLanguage() {
        return language;
    }

    public String getCopyright() {
        return copyright;
    }

    public String getPubDate() {
        return pubDate;
    }

    @Override
    public String toString() {
        return "Feed [copyright=" + copyright + ", description=" + description
                + ", language=" + language + ", link=" + link + ", pubDate="
                + pubDate + ", title=" + title + "]";
    }

}
```

Anexo 3: Clase Java denominada RSS*FeedParser* utilizada como base para el desarrollo de este TFG. Se hace referencia a este anexo en el capítulo de Implementación.

URL: https://www.vogella.com/tutorials/RSSFeed/article.html

```
package de.vogella.rss.read;

import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;

import javax.xml.stream.XMLEventReader;
import javax.xml.stream.XMLInputFactory;
import javax.xml.stream.XMLStreamException;
import javax.xml.stream.events.Characters;
import javax.xml.stream.events.XMLEvent;

import de.vogella.rss.model.Feed;
import de.vogella.rss.model.FeedMessage;

public class RSSFeedParser {
    static final String TITLE = "title";
    static final String DESCRIPTION = "description";
    static final String CHANNEL = "channel";
    static final String LANGUAGE = "language";
    static final String COPYRIGHT = "copyright";
    static final String LINK = "link";
    static final String AUTHOR = "author";
    static final String ITEM = "item";
    static final String PUB_DATE = "pubDate";
    static final String GUID = "guid";

    final URL url;

    public RSSFeedParser(String feedUrl) {
        try {
            this.url = new URL(feedUrl);
        } catch (MalformedURLException e) {
            throw new RuntimeException(e);
        }
    }

    public Feed readFeed() {
        Feed feed = null;
        try {
            boolean isFeedHeader = true;
            // Set header values intial to the empty string
            String description = "";
            String title = "";
            String link = "";
            String language = "";
            String copyright = "";
            String author = "";
            String pubdate = "";
            String guid = "";

            // First create a new XMLInputFactory
            XMLInputFactory inputFactory = XMLInputFactory.newInstance();
            // Setup a new eventReader
            InputStream in = read();
            XMLEventReader eventReader = inputFactory.createXMLEventReader(in);
            // read the XML document
            while (eventReader.hasNext()) {
                XMLEvent event = eventReader.nextEvent();
                if (event.isStartElement()) {
                    String localPart = event.asStartElement().getName()
                            .getLocalPart();
                    switch (localPart) {
                    case ITEM:
                        if (isFeedHeader) {
                            isFeedHeader = false;
                            feed = new Feed(title, link, description, language,
                                    copyright, pubdate);
                        }
                        event = eventReader.nextEvent();
                        break;
                    case TITLE:
                        title = getCharacterData(event, eventReader);
                        break;
                    case DESCRIPTION:
```

```
                        description = getCharacterData(event, eventReader);
                        break;
                    case LINK:
                        link = getCharacterData(event, eventReader);
                        break;
                    case GUID:
                        guid = getCharacterData(event, eventReader);
                        break;
                    case LANGUAGE:
                        language = getCharacterData(event, eventReader);
                        break;
                    case AUTHOR:
                        author = getCharacterData(event, eventReader);
                        break;
                    case PUB_DATE:
                        pubdate = getCharacterData(event, eventReader);
                        break;
                    case COPYRIGHT:
                        copyright = getCharacterData(event, eventReader);
                        break;
                    }
                } else if (event.isEndElement()) {
                    if (event.asEndElement().getName().getLocalPart() == (ITEM)) {
                        FeedMessage message = new FeedMessage();
                        message.setAuthor(author);
                        message.setDescription(description);
                        message.setGuid(guid);
                        message.setLink(link);
                        message.setTitle(title);
                        feed.getMessages().add(message);
                        event = eventReader.nextEvent();
                        continue;
                    }
                }
            }
        } catch (XMLStreamException e) {
            throw new RuntimeException(e);
        }
        return feed;
    }

    private String getCharacterData(XMLEvent event, XMLEventReader eventReader)
            throws XMLStreamException {
        String result = "";
        event = eventReader.nextEvent();
        if (event instanceof Characters) {
            result = event.asCharacters().getData();
        }
        return result;
    }

    private InputStream read() {
        try {
            return url.openStream();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

Anexo 4: Clase Java denominada *ReadTest* utilizada como base para el desarrollo de este TFG. Se hace referencia a este anexo en el capítulo de Implementación.

URL: https://www.vogella.com/tutorials/RSSFeed/article.html

```
package de.vogella.rss.tests;

import de.vogella.rss.model.Feed;
import de.vogella.rss.model.FeedMessage;
import de.vogella.rss.read.RSSFeedParser;

public class ReadTest {
    public static void main(String[] args) {
        RSSFeedParser parser = new RSSFeedParser(
                "https://www.vogella.com/article.rss");
        Feed feed = parser.readFeed();
        System.out.println(feed);
        for (FeedMessage message : feed.getMessages()) {
            System.out.println(message);

        }

    }
}
```

Anexo 5: Clase Java *Conexión* con la que se junta la base de datos MySQL con el programa en NetBeans.

```
package Conexion;

import com.mysql.jdbc.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

public class Conexion {

   private Connection conectar = null;

    public Connection hazConexion() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conectar                                =                                (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/tfg?autoReconnect=true&useSSL=false",  "root",
"bicicleta");
        } catch (ClassNotFoundException | SQLException error) {
            JOptionPane.showMessageDialog(null,    "Error    al    Conectarse"+"\n"+error,"Mensaje
Error",JOptionPane.ERROR_MESSAGE);
        }
        return conectar;

    }

    public void desconectar(){
        try {
            conectar.close();
        } catch (SQLException ex) {
            System.out.println("No se pudo desconectar");
        }
    }
}
```