



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

MAPA DE EMOCIONES EN DEBATES POLÍTICOS

Alumno: Andrea López Fernández

Tutoras: Prof. Dña. Salud M^a Jiménez Zafra
Prof. Dña. M^a Teresa Martín Valdivia

Dpto: Informática

Octubre, 2018



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Doña Salud M^a Jiménez Zafra y Doña M^a Teresa Martín Valdivia, tutoras del Proyecto Fin de Carrera titulado: Mapa de Emociones en Debates Políticos, que presenta Andrea López Fernández, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, OCTUBRE de 2018

La alumna:

Las tutoras:

Fdo: Andrea López Fernández

Fdo: Prof. Dña. Salud M^a Jiménez Zafra
Fdo: Prof. Dña. M^a Teresa Martín Valdivia

ÍNDICE

Resumen.....	7
1. INTRODUCCIÓN.....	8
1.1. Motivación.....	8
1.2. Propósito.....	9
1.3. Objetivos.....	10
1.4. Resultados esperados.....	10
1.5. Planificación temporal.....	11
1.6. Estimación de costes.....	12
1.6.1. Costes de hardware.....	12
1.6.2. Costes de software.....	12
1.6.3. Costes de personal.....	13
1.6.4. Coste total.....	13
1.7. Estructura del proyecto.....	14
2. ESTUDIO Y SELECCIÓN DE LA FUENTE DE DATOS.....	15
2.1. Fuente de datos seleccionada.....	15
2.2. Descarga de archivos de sesiones.....	16
2.3. Limpieza y almacenamiento de datos.....	17
2.3.1. Estructura de los archivos obtenidos.....	17
2.3.2. Almacenamiento.....	18
2.3.3. Limpieza.....	19
3. ESTUDIO DE TÉCNICAS PARA LA CLASIFICACIÓN DE EMOCIONES.....	20
3.1. Qué es una emoción.....	20
3.2. Modelos de emociones.....	21
3.2.1. Emociones de Ekman.....	22
3.3. Técnicas para clasificación de emociones.....	23
3.3.1. Técnica utilizada en el proyecto.....	23
3.4. Spanish Emotion Lexicon (SEL).....	24
4. DESARROLLO DEL PROYECTO.....	25
4.1. Descripción del problema.....	25
4.2. Objetivos del sistema.....	25
4.3. Especificación de requerimientos.....	26
4.3.1. Requisitos funcionales.....	26
4.3.2. Requisitos no funcionales.....	27
4.4. Metodología.....	27

4.4.1. Metodología ágil Kanban.....	28
4.5. Historias de usuario.....	29
4.6. Propuesta de solución.....	30
4.7. Justificación de la solución.....	31
4.8. Descripción de la solución.....	32
4.8.1. Iteración 1.....	32
4.8.2. Iteración 2.....	32
4.8.3. Iteración 3.....	33
4.8.4. Iteración 4.....	37
4.9. Herramientas para la implementación del sistema.....	38
4.9.1. Python.....	38
4.9.2. Flask.....	39
4.9.3. HTML, CSS, JavaScript y JQuery.....	39
4.9.4. Google Charts.....	40
5. CONCLUSIONES Y TRABAJOS FUTUROS.....	41
5.1. Conclusiones.....	41
5.2. Trabajos futuros.....	42
Bibliografía.....	43
ANEXO A. Manual de instalación.....	44
A.1. Instalación de Python 3.....	44
A.2. Instalación de pip.....	44
A.3. Instalación de paquetes de Python con pip.....	44
A.4. Ejecución del proyecto.....	45
ANEXO B. Manual de usuario.....	46
B.1. Búsqueda por sesión.....	47
B.1.1. Emociones en una sesión concreta.....	49
B.1.2. Emociones de un diputado en una sesión concreta.....	50
B.2. Búsqueda por diputado.....	51
B.2.1. Diputado concreto.....	52
ANEXO C. Índice de Ilustraciones.....	53
ANEXO D. Índice de Tablas.....	53

Resumen.

Este proyecto consiste en la creación de una aplicación web donde se podrán consultar las emociones que muestran los políticos del congreso de los diputados español, tanto en sesiones parlamentarias de un día concreto como los perfiles habituales de dichos diputados.

Antes de poder crear la aplicación web se ha hecho uso de técnicas de Procesamiento del Lenguaje Natural, haciendo especial hincapié en la tarea de clasificación de emociones en castellano, obteniendo así los datos que servirán de base para el sitio web.

1. INTRODUCCIÓN.

Las emociones están siempre presentes en el día a día de las personas y es un elemento intrínseco a nosotros, por lo que es un tema de interés para numerosos campos de estudio, tanto ciencia, como lingüística o incluso política. El poder provocar una emoción favorable a nuestras propuestas o controlar estas para prevenir enfermedades son algunos ejemplos.

En este proyecto la detección de estas emociones tendrá como propósito crear un mapa de emociones en los debates políticos que tienen lugar en el congreso de los diputados, ya que los políticos, como representantes que son, generan interés en la ciudadanía que los sigue y vigila que cumplan sus propuestas. Por tanto, un análisis emocional de estos puede ofrecer información interesante para sus votantes.

1.1. Motivación.

La detección y análisis de emociones de forma automática basándonos en los textos y el lenguaje ha sido un campo que ha ido prosperando cada vez más y que ha generado un alto interés en los últimos años, aunque aún le queda un largo recorrido. Es una ardua tarea que ha sido enfocada desde muchos puntos de vista, los cuales se explicarán más adelante en este mismo proyecto, y que engloba numerosas técnicas de Procesamiento del Lenguaje Natural (PLN), además de lingüística, procesamiento automático, psicología y un largo etcétera.

Este campo tiene numerosas aplicaciones: algo tan cotidiano como la comprobación de usabilidad de una web sabiendo qué emociones han experimentado los usuarios que la utilizan o como un apoyo en la ciencia de la psicología, con la generación de perfiles emocionales para la detección de enfermedades, siendo útil para el tratamiento de nuestra salud mental. También se puede utilizar para generar sondeos políticos en unas elecciones electorales y saber qué actitudes y propuestas hacen que los ciudadanos tengan mejores respuestas emocionales a los candidatos propuestos.

Sin embargo, en este caso se ha enfocado este trabajo desde la perspectiva contraria: esta vez se analizará a los políticos del congreso de los diputados a través de las publicaciones que se actualizan periódicamente en su respectivo sitio web. Gracias a estos documentos se podrán extraer los fragmentos de texto relativos a cada uno de los diputados y se podrán analizar dichos textos para obtener varias estadísticas y perfiles emocionales.

Además de ser un tema curioso, es un trabajo realizado en castellano, lo cual también se distingue de otro tipo de aplicaciones dedicadas a este tema, ya que la mayoría de investigaciones y estudios enfocados al análisis de emociones están realizados en inglés, por lo que el idioma también supondrá un reto.

1.2. Propósito.

El propósito de este trabajo será la investigación de un área específica del Procesamiento del Lenguaje Natural, siendo esta la extracción de emociones basándonos solamente en el texto, es decir, en las palabras y expresiones utilizadas. Como ya se ha mencionado anteriormente, existen numerosos trabajos enfocados al inglés. Sin embargo no ocurre lo mismo en el castellano, por lo que esta aplicación se desarrollará en dicho idioma.

Además, la idea de generar un mapa de emociones políticas nos ayudará a entender mejor a nuestros representantes en el parlamento. Podremos ver cuál es su estado emocional general y su variabilidad a lo largo del tiempo, además de información específica según el día de una sesión concreta. Es un tema innovador que pretende extraer conocimiento de este tipo de datos.

1.3. Objetivos.

Los objetivos de este proyecto son los siguientes:

- Estudiar el portal web del congreso de los diputados para seleccionar la información con la que se trabajará.
- Estudiar las distintas aproximaciones que hay actualmente para la detección y reconocimiento de emociones.
- Seleccionar las técnicas más adecuadas para el dominio concreto de aplicación.
- Desarrollar una aplicación web que implemente sistemas de detección y reconocimiento de emociones mostrando un mapa de dichas emociones.
- Redactar una memoria que recoja todo el trabajo desarrollado, así como los manuales de instalación y de usuario.

1.4. Resultados esperados.

Los resultados que se desean obtener con este proyecto son los siguientes:

- Estudio y comparación de los diferentes recursos que se utilizan para la detección de emociones.
- Recopilación automática de fuentes de datos adecuadas, es decir, los textos de las sesiones.
- Implementación de un programa que obtendrá los datos relacionados con las emociones, aplicando la técnica seleccionada.
- Una interfaz web que hará uso de los datos obtenidos anteriormente.
- Memoria del proyecto.
- Manuales de instalación y de usuario.

1.5. Planificación temporal.

El proyecto se ha dividido en una serie de tareas que se han ido desarrollando y organizado en diferentes intervalos de tiempo. Esta información se detallará en este apartado, donde se mostrará una tabla con las distintas actividades realizadas además de su respectivo diagrama de Gantt, que nos ayuda a ver el progreso de una forma más sencilla e intuitiva.

A continuación se puede ver detallada la tabla de tareas (Tabla 1.1), con su respectiva duración expresada en días y sus correspondientes fechas de inicio y fin.

Nombre de la tarea	Duración	Comienzo	Fin
Estudio y selección de fuente de información	8 días	12/02/2018	20/02/2018
Recopilación de documentos	16 días	21/02/2018	09/03/2018
Estudio de técnicas de reconocimiento de emociones	18 días	10/03/2018	28/03/2018
Almacenamiento de información	17 días	29/03/2018	15/04/2018
Aplicación técnicas de reconocimiento de emociones	24 días	16/04/2018	10/05/2018
Estudio de tecnologías para la aplicación web	9 días	11/05/2018	20/05/2018
Implementación de la web	116 días	21/05/2018	14/09/2018
Pruebas de usuario	20 días	15/09/2018	05/10/2018
Manuales de instalación y de usuario	9 días	06/10/2018	15/10/2018
Generación de una memoria	255 días	12/02/2018	25/10/2018

Tabla 1.1. Planificación temporal del proyecto.

Como se puede observar, el proyecto ha tenido una duración total de 255 días, dando comienzo el 12 de febrero y finalizando el 25 de octubre de este mismo año.

Además, se complementa la información relativa a la planificación con un diagrama de Gantt (Ilustración 1.1), que ayudará a ver de forma más sencilla la división de tareas a lo largo del tiempo.

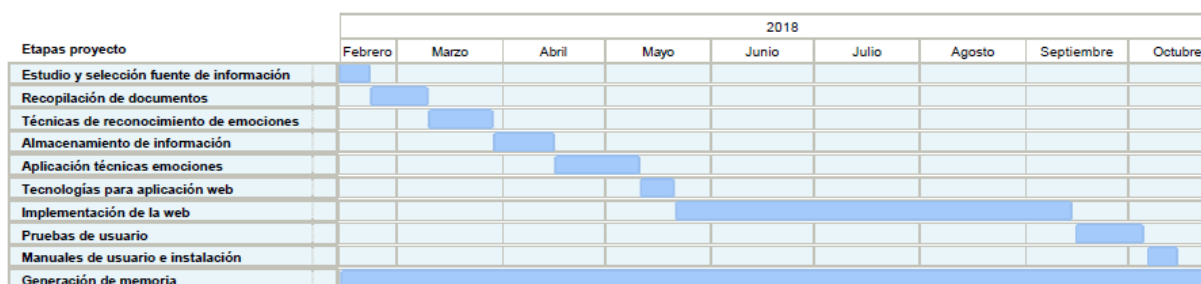


Ilustración 1.1. Diagrama de Gantt.

1.6. Estimación de costes.

Todo proyecto de desarrollo de software conlleva unos costes asociados a su desarrollo y mantenimiento, por eso es importante desglosarlos en este apartado.

1.6.1. Costes de hardware.

Los costes del hardware están relacionados con las herramientas físicas utilizadas para el desarrollo del proyecto. En este caso solo es necesario un ordenador, que aunque puede ser uno equivalente, en este proyecto el que ha sido utilizado es el siguiente:

Ordenador portátil HP 250 G2 Notebook PC, que cuenta con:

- Procesador: Intel Core i5-3230M (2.6GHz)
- Memoria RAM: 4GB.
- Memoria de disco: 500GB.
- Tarjeta gráfica: Intel HD Graphics 4000.

El coste inicial del hardware utilizado sería de 400 euros, sin embargo, se estima que este equipo tiene una vida útil de cuatro años y debido a que este fue adquirido en febrero de 2014, se concluye que el coste total de hardware no será significativo para este proyecto. Por tanto, el coste total de hardware es: 0€.

1.6.2. Costes de software.

Los costes de software son los relativos a los programas utilizados para la realización del proyecto, es decir, las licencias necesarias para poder obtenerlos. El software utilizado es el siguiente:

Windows 10, incluido en la compra del PC	0€
Pycharm Community	0€
Google Drive	0€
Mozilla Firefox y Google Chrome	0€

Se concluye que el coste total relativo al software tampoco será significativo debido a que las licencias son gratuitas, es decir, el coste será de 0 euros.

1.6.3. Costes de personal.

Este será el coste de las personas que se han dedicado a desarrollar el proyecto. En este caso se considerarán tres puestos de trabajo:

- **Analista:** la persona encargada del desarrollo de algoritmos y del diseño del sistema, comparando y comprobando los resultados obtenidos. En este caso sería alguien especializado en Procesamiento del Lenguaje Natural y en el análisis de emociones.
- **Programador:** será quién se encargue de implementar el programa diseñado por el analista en el paso anterior.
- **Desarrollador web:** por último, la persona encargada del diseño e implementación de la interfaz web.

En la Tabla 1.2 se muestra el despliegue de gasto dirigido al personal del proyecto, siendo los sueldos tomados desde [indeed](https://www.indeed.es/)¹:

Puesto	Sueldo mensual	Nº de meses	Total
Analista	1760€	7 meses	12300€
Programador	1200€	3 meses	3600€
Desarrollador web	1500€	4 meses	6000€

Tabla 1.2. Coste de personal.

1.6.4. Coste total.

Finalmente obtenemos el coste total del proyecto, siendo este la suma del coste hardware, coste software, coste de personal y el porcentaje de beneficio que se desea obtener, que en este caso se ha decidido que sea de un 15%.

De este modo, en la siguiente tabla (Tabla 1.3) se muestra el desglose total de gastos.

Tipo de coste	Gasto
Hardware	0€
Software	0€
Personal	21900€
Suma de costes	21900€
Beneficio del 15%	3285€
TOTAL	25185€

Tabla 1.3. Coste total del proyecto.

¹ <https://www.indeed.es/> - Portal especializado en empleo.

1.7. Estructura del proyecto.

En este apartado se describirá la estructura de la memoria correspondiente al proyecto, la cual está dividida en cinco capítulos, la bibliografía y cuatro anexos. Se procede a la descripción de dichos apartados:

- **Capítulo 1.** Es la introducción del proyecto. Aquí se explica la motivación además de los propósitos y objetivos que se pretenden alcanzar. Además, se hace un desglose de la planificación temporal y de gastos, que son necesarios para un buen planteamiento.
- **Capítulo 2.** Aquí se explicará cuáles son los datos que se han cogido para este proyecto, cómo se han obtenido y cómo se han almacenado para que estos estén listos a la hora de aplicar las técnicas de extracción de emociones.
- **Capítulo 3.** Es el capítulo donde se hará un estudio y una comparación de las distintas metodologías existentes a la hora de la clasificación y extracción de emociones mediante texto. Finalmente se expondrá la solución adoptada y los motivos.
- **Capítulo 4.** Es el apartado que explica cómo se ha desarrollado el proyecto. Este incluye la selección de la metodología y la descripción detallada de todos los pasos dados hasta la obtención del resultado final.
- **Capítulo 5.** Este último apartado incluye las conclusiones obtenidas a partir del trabajo realizado, además de posibles mejoras y trabajos futuros.
- **BIBLIOGRAFÍA.** Aquí se encuentran las referencias y recursos consultados para la realización del proyecto.
- **ANEXO A.** El manual de instalación, donde se explicará paso por paso los elementos necesarios para poder instalar la aplicación desarrollada.
- **ANEXO B.** Manual de usuario, que como indica su nombre, es una guía de uso de la aplicación destinada a los usuarios que vayan a utilizarla.
- **ANEXOS C y D.** Contienen los índices de tablas y de ilustraciones respectivamente.

2. ESTUDIO Y SELECCIÓN DE LA FUENTE DE DATOS.

En este apartado se describirá la fuente de datos utilizada, además del tratamiento seguido para que estos estén listos para desarrollar este proyecto.

2.1. Fuente de datos seleccionada.

Este proyecto es posible gracias a la recopilación que existe de todas las intervenciones relativas a las sesiones parlamentarias en el [sitio web del congreso](#)² de los diputados. Los documentos están clasificados por legislatura, año y mes, lo que hace muy sencilla la localización de una sesión concreta. En la Ilustración 2.1 es posible ver la organización del sitio web, observándose en este ejemplo las sesiones del mes de julio de 2018.

The screenshot shows the website of the Congreso de los Diputados. At the top left is the logo with the text 'Congreso de los Diputados' and 'constitución'. To the right is a navigation menu with categories like 'Actualidad', 'Sala de Prensa', 'Congreso TV', 'Diputados', 'Grupos', 'Organos', 'Información', 'Catálogo de Publicaciones', 'Iniciativas', 'Intervenciones', 'Historia y Normas', 'Edición oficial', 'Actividad', 'Internacional', 'Servicios Documentales', and 'Unión Europea'. Below the menu is a search bar and a list of sessions for July 2018, including dates like 02/07/2018, 04/07/2018, 10/07/2018, 16/07/2018, 17/07/2018, 25/07/2018, and 27/07/2018, each with a PDF link. At the bottom, there is a footer with contact information and social media links.

Ilustración 2.1. Web del congreso de los diputados.

Sabiendo esto se ha tomado la primera decisión: este trabajo estará enfocado en la XII legislatura, es decir, la legislatura que transcurre desde 2016 hasta nuestros días, puesto que al ser una idea innovadora se ha asumido que el enfoque más interesante sería uno actual que incluya a los políticos que nos representan hoy en día, de los cuales se habrán de obtener perfiles emocionales.

²

http://www.congreso.es/portal/page/portal/Congreso/Congreso/Publicaciones/DiaSes/Pleno?_piref73_1340168_73_1340165_1340165.next_page=/wc/diarioSesiones&anioDiarios=2018&tipoDiario=1&idLegislatura=12

2.2. Descarga de archivos de sesiones.

Cada una de las sesiones de esta legislatura concreta tiene un enlace correspondiente que nos lleva a archivos de extensión “.pdf”, donde estarán redactadas las intervenciones según el día seleccionado. Estos archivos tienen urls del tipo:

“[http://www.congreso.es/public_oficiales/L12/CONG/DS/PL/DSCD-12-PL-\[numsesion\].PDF](http://www.congreso.es/public_oficiales/L12/CONG/DS/PL/DSCD-12-PL-[numsesion].PDF)”

Si han sufrido alguna corrección, se añadirá a este mismo enlace “-C1” antes de la extensión de pdf. También dicha extensión puede aparecer en minúscula, pero en general el patrón se repite. Las sesiones están numeradas en orden ascendente, de más antiguo a más reciente, y esto viene indicado en “numsesion”, por lo que se usará esta uniformidad de urls para obtener los archivos que nos interesan. Toda esta información se puede ver de forma más clara en la Ilustración 2.2.

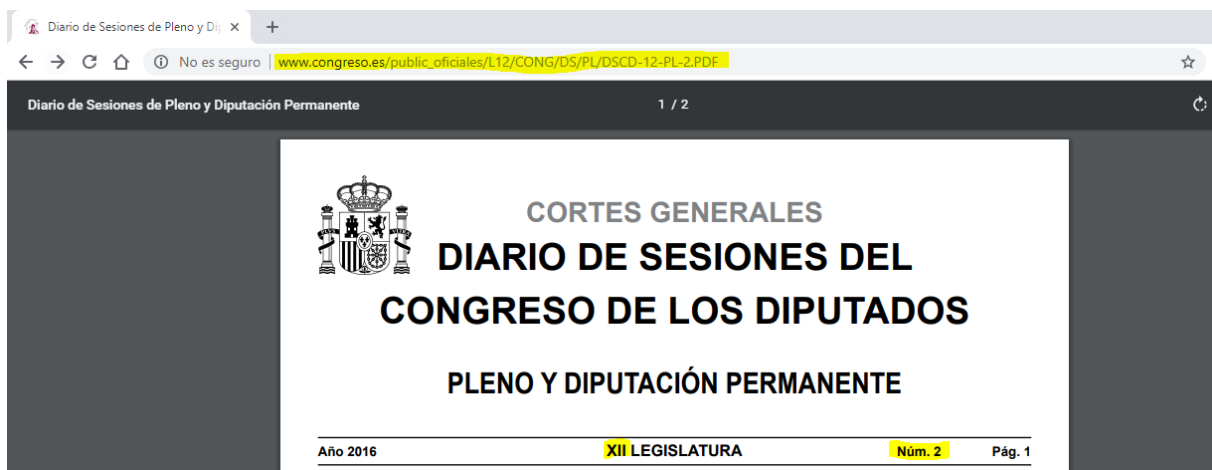


Ilustración 2.2. Ejemplo de url de pdf.

Teniendo toda esta información en cuenta, se concluye que la manera más sencilla y rápida de obtener los archivos de texto será mediante la implementación de un crawler que los descargue automáticamente, siguiendo las pautas descritas en este mismo apartado para crearlo.

Un crawler es un programa cuyo objetivo es la descarga automática de ciertos elementos que nos interesen de una web. Hay de muchos tipos, desde los generales que obtienen cada uno de los elementos que conforman las páginas de un sitio web cualquiera hasta los más específicos que buscan componentes de un determinado tipo en una web concreta. El programa desarrollado en este proyecto será del segundo tipo, siendo este diseñado solo para el propósito que nos interesa.

2.3. Limpieza y almacenamiento de datos.

Una vez se ha escogido la fuente de datos y se ha elegido la técnica para recopilarlos, el siguiente paso será la limpieza y almacenamiento, dejando dichos datos listos para la tarea de análisis de emociones.

Es importante que nuestra fuente de información esté bien preparada para el uso específico que se hará a posteriori y para que el acceso a esta sea eficaz y eficiente. Por ello habrá que separar los datos que no son útiles de los que sí lo son, desarrollando un algoritmo adaptado a este problema para solo quedarnos con estos últimos.

2.3.1. Estructura de los archivos obtenidos.

Como hemos mencionado, el algoritmo debe estar adaptado a nuestro problema y para que esto sea así se deben analizar los archivos que se han descargado y buscar patrones que nos ayuden a guardar de forma efectiva las intervenciones de cada diputado. En la Ilustración 2.3 se puede ver un extracto de una sesión aleatoria.

La señora **PRESIDENTA**: Señorías, por favor, guarden silencio.

El señor **VENDRELL GARDENES**: ¿Es que a nosotros no nos preocupa el déficit y la deuda? Claro que nos preocupa. ¿Qué deuda nos preocupa? Los ingentes recursos destinados al rescate de las autopistas, a las estaciones de alta velocidad, al Castor, los 3.500 millones de la estafa eléctrica de los costes de transición a la competencia, los 26.300 millones de ayudas a la banca que no volveremos a ver más... y la corrupción, señores del Partido Popular. La corrupción también genera deuda porque con una mano ustedes han recortado servicios públicos y con otra han amparado la corrupción. Y nos preocupa otro tipo de déficit: el déficit social. El hecho de que en España, por ejemplo, 1.700.000 personas más son pobres en relación con el año 2009; o la pobreza infantil, que genera deuda social y traslada la desigualdad y la exclusión a las generaciones futuras.

La desigualdad crece con el desempleo, la precariedad y por la debilidad de las políticas públicas. ¿Cuál es nuestro problema? Que ingresamos ocho puntos menos de la media europea y somos el segundo país de la Unión Europea, después de Irlanda, con menos gasto público. Ese es el problema. Las políticas de austeridad son el eje del proyecto ideológico del Partido Popular: menos Estado del bienestar y más desigualdad. Pero, eso sí, más Estado centralizado. No podemos desaprovechar esta ocasión para realizar un cambio profundo de la ley que ponga fin a las políticas de austeridad injustas y recentralizadoras. Poner fin de una vez por todas a la hegemonía de la austeridad.

Gracias. **(Aplausos)**.

La señora **PRESIDENTA**: Gracias.
Tiene la palabra el señor Montero Soler.

El señor **MONTERO SOLER**: Buenas tardes, señorías.

Hago más las consideraciones que han hecho los miembros de mi grupo parlamentario que me han precedido en el uso de la palabra y plantearé algunas cuestiones de forma muy sintética por el tiempo tan restringido.

Ilustración 2.3. Extracto de pdf de sesión.

En la ilustración anterior se puede observar un patrón regular a la hora de mostrar cada una de las intervenciones de los diputados del congreso de los diputados. Este patrón tiene la forma:

- *“El señor/la señora” más un nombre en mayúscula y en negrita.*

Además, los archivos tienen dos oraciones específicas para el comienzo de cada sesión, haciéndose uso de solo una de ellas según el caso. Estas son:

- *“Se abre la sesión...”*
- *“Se reanuda la sesión...”*

La fórmula de cierre siempre es la última oración, que indica la hora de término de la sesión. Además, la fecha de cada sesión viene especificada en cada página de los pdf.

Por tanto, con toda esta información será sencillo deshacernos de la información innecesaria (como los sumarios escritos al inicio de cada archivo) y quedarnos solo con las palabras pronunciadas por los políticos de la cámara baja.

2.3.2. Almacenamiento.

Existen numerosas estructuras de almacenamiento, pero en este caso se usarán diccionarios para manejar los datos y para almacenarlos de forma estática se usará la librería “numpy” para Python, que nos permite abrirlas y guardarlos de forma sencilla siempre que lo necesitemos

Un diccionario o mapa es una estructura de datos que contiene una lista de claves, y a su vez, asociados a cada clave, los datos que se quieran introducir, desde una sola variable hasta otra estructura. Por lo tanto, una vez se haga referencia a una de las claves, la estructura devolverá los datos asociados a esta. En la Ilustración 2.4 se muestra un pequeño esquema.

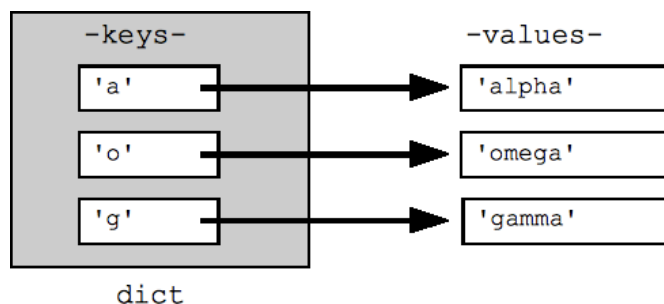


Ilustración 2.4. Estructura de un diccionario.

Se cree conveniente esta estructura porque se adecúa perfectamente a la estructura de los datos que se quieren extraer. Puesto que el objetivo de los datos es el uso de estos en una web, se prioriza que el acceso a la información que se requiera en esta sea rápido y este es un hecho que el diccionario cumple a la perfección, siendo este tiempo casi imperceptible.

Por tanto se quiere obtener como resultado final un diccionario de diccionarios que finalmente contendrá las intervenciones. En otras palabras, tendremos:

- Fecha de sesión: primera clave que nos lleva a otro diccionario con los diputados que han intervenido en ese día concreto.
- Diputado: segunda clave que nos lleva a una lista de intervenciones de tipo string.
- Lista de tipo string: texto final de las intervenciones.

2.3.3. Limpieza.

Ahora que tenemos los datos organizados, hay que guardarlos de la forma más eficiente posible, es decir, hay que quedarse solo con lo necesario para que consuma la menor cantidad de memoria posible. Hay que tener en cuenta que en este caso se trabajará con muchísimo texto por lo que este paso es esencial para el correcto funcionamiento del proyecto.

Seguimos dos pasos: eliminación de “palabras vacías” y “lematización”.

1. **Stopwords.** las stopwords o palabras vacías son aquellas palabras que carecen de significado semántico y que no aportan ninguna información. Un ejemplo de esto son los artículos o las preposiciones. Se eliminarán estas palabras para que el uso de memoria sea más eficiente.
2. **Lematización.** Un lema es la unidad mínima con significado asociada a una palabra específica. Por ejemplo, el significado de “habló” y “hablando” es el mismo, por lo que se podría reducir a una sola forma: “hablar”. Esto nos permitirá detectar el significado de las palabras asociadas a las emociones independientemente de su forma verbal o de los prefijos o sufijos que esta lleve.

3. ESTUDIO DE TÉCNICAS PARA LA CLASIFICACIÓN DE EMOCIONES.

En este apartado se exponen las distintas técnicas y modelos existentes para la clasificación de emociones, explicándose cuáles de ellos serán los utilizados para este proyecto.

3.1. Qué es una emoción.

Según la segunda acepción de la RAE, una emoción se define como: “Interés, generalmente expectante, con que se participa en algo que está ocurriendo” (Ilustración 3.1.).

emoción

Del lat. *emotio*, *-ōnis*.

1. f. Alteración del ánimo intensa y pasajera, agradable o penosa, que va acompañada de cierta conmoción somática.
2. f. Interés, generalmente expectante, con que se participa en algo que está ocurriendo.

Real Academia Española © Todos los derechos reservados

Ilustración 3.1. Definición de "emoción" en la RAE.

También se puede definir como “una respuesta discreta y consistente hacia eventos tanto externos como internos relativos a una persona, los cuales tienen una cierta relevancia para esta; además, tienen una duración corta” (Yadollahi, Gholipour Shahraki, & R. Zaiane, 2017). En otras palabras, son las reacciones que tiene el ser humano cuando percibe un objeto, lugar, suceso u otra persona. Algunos ejemplos de emociones son alegría, tristeza o sorpresa. Cuando se dice que tienen “cierta relevancia” para la persona quiere decir que tiene una repercusión en su sistema biológico (como en sus expresiones faciales).

Este tipo de reacciones, las emociones, pertenecientes a los políticos, son las que se tratarán de percibir en la aplicación desarrollada, basándonos en el texto de las sesiones parlamentarias recopilado previamente.

3.2. Modelos de emociones.

El estudio de la clasificación de emociones y de la semántica que esta implica dio comienzo en la década de 1960 (Stone, Dunphy, Smith, & Ogilvie, 1968). Por este motivo, existen numerosas teorías en cuanto a los modelos de emociones existentes.

Estos modelos se pueden clasificar en modelos de emociones discretas o de emociones dimensionales. El modelo discreto afirma que cada emoción proviene de un sistema neuronal diferente, mientras que el dimensional expone que existe un sistema neuropsicológico que es responsable de todo tipo de emociones (Yadollahi, Gholipour Shahraki, & R. Zaiane, 2017).

Atendiendo a esta clasificación, diferentes investigadores han propuesto diferentes conjuntos de emociones que para ellos son las básicas. Varios de estos modelos se pueden ver en la Tabla 3.1 (Yadollahi, Gholipour Shahraki, & R. Zaiane, 2017), siendo estos propuestos por (Ekman, Friesen, & Ellsworth, 1972), (Plutchik & Kellerman, 1986), (Shaver, Schwartz, Kirson, & O'connor, 1987) y (Lövheim, 2012).

Emoción	Ekman (discreto)	Plutchik (dimensional)	Shaver (discreto)	Lövheim (dimensional)
Enfado	X	X	X	X
Anticipación		X		
Asco	X	X		X
Angustia				X
Miedo	X	X	X	X
Interés				X
Alegría	X	X	X	X
Amor			X	
Tristeza	X	X	X	
Vergüenza				X
Sorpresa	X	X	X	X
Confianza		X		

Tabla 3.1. Modelos de emociones.

3.2.1. Emociones de Ekman.

Paul Ekman fue un reconocido psicólogo que encabezó un experimento cuyo objetivo era el análisis de la conducta no verbal y de las expresiones faciales asociadas a cada emoción. Tras este estudio, Ekman concluyó que las emociones básicas serían seis: enfado, miedo, asco, sorpresa, alegría y tristeza (Ilustración 3.2).

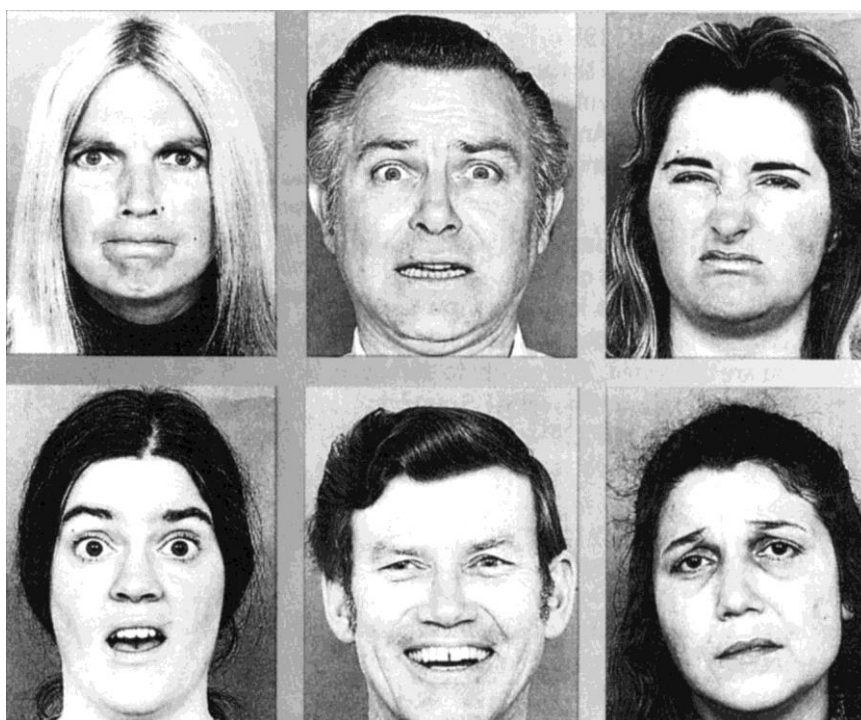


Ilustración 3.2. Emociones de Ekman.

A pesar de basarse en lenguaje no verbal, este conjunto de emociones ha sido ampliamente utilizado para clasificar los textos según la emoción implícita en ellos.

Debido a la sencillez del modelo y a la cantidad de recursos que existen basados en él, este ha sido el elegido para crear los perfiles emocionales de los políticos del congreso de los diputados.

3.3. Técnicas para clasificación de emociones.

Actualmente las técnicas para clasificar emociones se pueden dividir en dos grandes grupos: aquellas basadas en el léxico y las que utilizan aprendizaje automático.

Las técnicas que se basan en el léxico pueden hacer uso de lexicones, bolsas de palabras (bags of words) u ontologías (Canales & Martínez-Barco, 2014), es decir, se basan en recursos lingüísticos.

Los lexicones son conjuntos de palabras donde cada una de ellas está clasificada con emoción específica. Utilizando este recurso es posible clasificar un texto según sus emociones, comparando el vocabulario nuevo con el existente en el lexicón. Una bolsa de palabras se basa en el mismo concepto, pero no tiene en cuenta el orden de aparición de los términos.

Las ontologías se basan en conceptos, es decir, en las definiciones de las palabras. Por ejemplo, “alegría” y “felicidad” estarían clasificadas en el mismo concepto.

Por otro lado tenemos el aprendizaje automático. Este método hace uso de algoritmos y de conjuntos de datos de entrenamiento para construir un modelo que permita clasificar nuevos elementos.

Por ejemplo, si lo que queremos es clasificar emociones presentes en tweets habría que buscar o generar un corpus etiquetado, es decir, un conjunto de tweets ya correctamente clasificados. Entonces, haciendo uso de uno de los numerosos algoritmos que existen, se construiría un modelo basado en este corpus el cual se divide en dos: una parte para entrenar el modelo y otra para comprobar la efectividad del mismo, obteniendo un porcentaje de acierto para el mismo. Para clasificar los nuevos tweets solo hay que aplicarles el modelo construido.

Además existe otra técnica: el modelo híbrido, que uniría los dos explicados previamente.

3.3.1. Técnica utilizada en el proyecto.

Tras analizar las técnicas empleadas en este tipo de tarea, se ha optado por la utilización de un lexicón. Esto es así debido a la naturaleza de este proyecto, ya que al ser una idea sobre la que no se ha trabajado antes no existen corpus de este dominio político anotados con emoción en castellano y por tanto no tenemos una base para crear un modelo basado en aprendizaje automático o un modelo híbrido. Por tanto, el uso de un lexicón general de emociones es la mejor opción.

3.4. Spanish Emotion Lexicon (SEL).

Por último, tras escoger el modelo de emociones (las seis emociones de Ekman) y la técnica que se usará para su clasificación (uso de un lexicón), hay que buscar un recurso apropiado para la aplicación que se desarrollará.

Puesto que el número de recursos en castellano es limitado, y menos aún con el modelo específico seleccionado, se ha optado por utilizar un lexicón ya desarrollado previamente en castellano. Este es el “Spanish Emotion Lexicon” o SEL (Díaz Rangel, Sidorov, & Suárez-Guerra, 2014).

Las palabras de este recurso están clasificadas en seis emociones básicas: alegría, enojo, miedo, repulsión, sorpresa y tristeza. Para su creación se tomó como base el recurso en inglés WordNet-Affect (Strapparava & Valiutti, 2004) y se tradujo automáticamente al castellano mediante los traductores de Google, English-Spanish Interpreter Pro y Babylon, haciendo una posterior limpieza de duplicados para que el resultado fuera consistente. Tras un análisis manual de las palabras traducidas, el número total de términos obtenidos es de 2036. Un desglose más detallado se puede ver en la Tabla 3.2 (Díaz Rangel, Sidorov, & Suárez-Guerra, 2014).

CATEGORÍA (emoción)	Palabras en español
Alegría	668
Repulsión	209
Enojo	382
Miedo	211
Sorpresa	175
Tristeza	391
TOTAL	2036

Tabla 3.2. Número de palabras del lexicón SEL.

Este es un lexicón robusto, que ha sido probado en numerosas aplicaciones y que tiene muy buenos resultados (Sidorov, y otros, 2012) por lo que se selecciona como la mejor opción para el proyecto actual.

4. DESARROLLO DEL PROYECTO.

En este apartado se explicará paso a paso cómo ha sido el desarrollo del proyecto, desde el planteamiento hasta la implementación de este.

4.1. Descripción del problema.

Los políticos que trabajan en el congreso de los diputados son personas importantes cuyas decisiones afectan a la ciudadanía, por tanto es interesante analizar sus comportamientos y más aún, sus emociones.

Por ello, surge la idea de crear un mapa de emociones en debates políticos basándonos en los textos transcritos de sus intervenciones en el congreso de los diputados y haciendo uso de técnicas destinadas al análisis de emociones, centrándonos en el castellano como idioma base.

4.2. Objetivos del sistema.

El objetivo principal del sistema es crear una herramienta que permita a la ciudadanía consultar estadísticas y gráficas relacionadas con los perfiles emocionales que se muestran en los debates políticos. Esta información debe presentarse en una interfaz atractiva e intuitiva para el usuario, de manera que pueda acceder a ella fácilmente, tanto si las emociones que quiere consultar son de un político en concreto como si quiere saber lo que pasó en una sesión cualquiera.

Para ello, el sistema deberá:

- Mostrar los días en los que se ha celebrado una sesión en el congreso y dar la posibilidad de elegir cualquiera de estos.
- Mostrar una lista de los políticos que han intervenido en una sesión concreta.
- Mostrar una lista de todos los políticos que componen la cámara baja.
- Analizar y mostrar los perfiles emocionales para cualquiera de las tres opciones anteriores.

4.3. Especificación de requerimientos.

Los requerimientos se refieren al comportamiento que va a tener el sistema que se va a desarrollar. Estos se pueden dividir en requisitos funcionales y no funcionales, los cuales serán descritos a continuación.

4.3.1. Requisitos funcionales.

Los requisitos funcionales describen cada una de las funciones que el sistema debe realizar y en el actual proyecto son los siguientes:

- La aplicación debe tener los datos relativos a las emociones disponibles, es decir, que la información esté almacenada de forma estática. [R.F.1]
- La aplicación debe contener dos partes claramente diferenciadas: diputados y sesiones. [R.F.2]
- La aplicación debe contener una lista de diputados disponibles, siendo posible acceder a cada perfil individual. En este se podrán observar las gráficas relativas a sus emociones habituales. [R.F.3]
- En la aplicación se debe poder elegir una fecha en la que haya habido una sesión parlamentaria y encontrar perfiles emocionales relativos a ese día concreto. [R.F.4]
- Una vez elegida una fecha se debe poder ver qué diputados han comparecido ese día y poder ver para cada uno de ellos el perfil emocional que han mostrado en esa sesión concreta. [R.F.5]

4.3.2. Requisitos no funcionales.

Los requisitos no funcionales describen una serie de propiedades que añaden restricciones tanto en el diseño como en la implementación del producto y estos son:

- La muestra de gráficas y de los datos de interés debe ser eficiente, es decir, debe tardar el menor tiempo posible (un máximo de un segundo) para que el usuario no espere más tiempo del necesario. [R.N.F.1]
- La aplicación debe contemplar todos los casos de uso, añadiendo excepciones y mostrando errores si fuera necesario. [R.N.F.2]
- El sitio web obtenido debe tener en cuenta los criterios de diseño y usabilidad, siendo este intuitivo y accesible para todo el mundo. [R.N.F.3]
- La aplicación debe funcionar en los navegadores más comunes (Google Chrome, Firefox, Microsoft Explorer o Edge y Safari). [R.N.F.4]
- La aplicación web debe ser responsive, es decir, se ha de adaptar a diferentes dispositivos para su fácil visualización.

4.4. Metodología.

La metodología es un paso primordial en todo desarrollo de un proyecto software ya que permite seguir una disciplina de trabajo, estructurando, planificando y controlando este proceso de desarrollo. Con este propósito existen numerosos tipos de metodologías, cada una con sus propias ventajas e inconvenientes y con un marco de trabajo diferente.

En este caso se ha optado por la denominada “metodología ágil”, la cual nos describe un desarrollo iterativo e incremental. En otras palabras, esta metodología propone la división del proyecto en varias subtarefas, las cuales se tratarán como si fueran proyectos individuales, con sus correspondientes análisis, diseño, desarrollo, prueba e implementación (Ilustración 4.1).

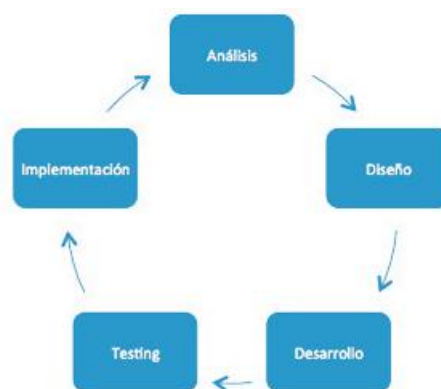


Ilustración 4.1. Etapas en cada iteración de la metodología ágil.

Sin embargo, el paso más destacado de este método es la revisión o reunión con el cliente, el cual podrá ir comprobando al cabo de cada iteración el avance del proyecto. Esto permitirá realizar correcciones sobre lo anterior y adaptarlo a lo que el cliente nos diga, pudiendo avanzar a la siguiente iteración solo cuando haya la seguridad de que lo hecho hasta ese momento es correcto.

Debido a la naturaleza de este proyecto y a la posibilidad de reunión con el cliente de forma habitual (en este caso, el tutor), se ha optado por esta opción, ya que es posible ir corrigiendo lo que se vaya desarrollando conforme avance el trabajo.

4.4.1. Metodología ágil Kanban.

Dentro de la metodología ágil existen muchos tipos y de entre todos ellos se ha utilizado la metodología denominada “Kanban” (Ilustración 4.2). Kanban establece que solo se proseguirá con un trabajo si existe la posibilidad de realizarlo (tanto por recursos como por tiempo) y sigue una estructura organizada por tarjetas. La principal ventaja es que es una metodología adaptativa, de las que menos resistencia al cambio presenta y por eso ha sido la seleccionada.

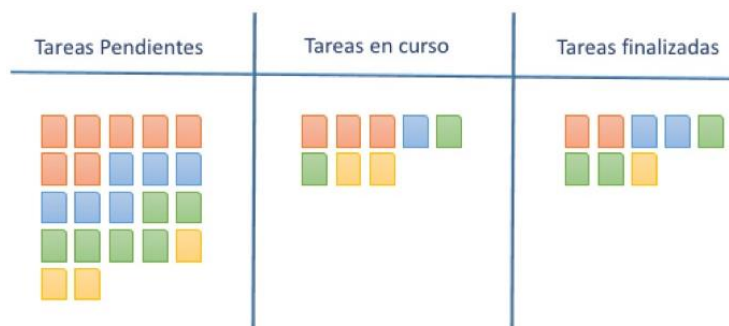


Ilustración 4.2. Metodología Kanban.

4.5. Historias de usuario.

Las historias de usuario permiten describir lo que ha de construirse en un proyecto software, asociándose a estas una prioridad y el tiempo estimado para su realización.

Estas historias se dividen en tareas y cada una de ellas incluye una breve descripción que permitirá hacer un primer esbozo de lo que se quiere realizar, ya que, como se ha explicado en el capítulo anterior, esta metodología está sujeta a cambios constantes.

En la siguiente tabla (Tabla 4.1) se podrán ver estas historias (entre paréntesis se muestran las dependencias entre las mismas).

ID (dep)	Nombre	Prioridad	Descripción
1	Descarga de archivos	Alta	Como usuario quiero disponer de una base de trabajo con los diferentes documentos de texto del congreso de los diputados.
2 (1)	Lectura de archivos	Alta	Como usuario quiero que el texto sobre el que se trabaja esté bien estructurado y almacenado para su posterior uso.
3 (2)	Aplicación de técnicas de clasificación de emociones	Alta	Como usuario quiero que se extraigan las emociones de la mejor manera posible.
4 (3)	Creación de interfaz web	Alta	Como usuario quiero disponer de una interfaz para poder interactuar con la aplicación creada de forma fácil e intuitiva.

Tabla 4.1. Historias de usuario.

4.6. Propuesta de solución.

Aquí se explicará la solución adoptada, enumerando los pasos a seguir hasta llegar al resultado final: la interfaz web.

1. Lo primero que hay que hacer es descargar los archivos sobre los que se basará el análisis de emociones y por tanto, la aplicación web. Para ello se procederá al análisis de la web del congreso de los diputados y a la creación de un crawler que recopile todos los archivos “pdf” relativos a las sesiones de la XII legislatura, tal y como se explicó en el apartado 2.2.
2. El siguiente paso es analizar la estructura de los pdfs y extraer solo la fecha de la sesión, los diputados que han hablado y sus respectivas intervenciones, limpiando el texto para dejarlo preparado para su posterior análisis.
3. Crear un programa que aplique las técnicas de extracción de emociones seleccionadas y obtener los datos resultantes con los perfiles emocionales correspondientes.
4. Por último, crear una interfaz web que haga uso de los datos obtenidos en el paso anterior. Se propone una interfaz que contenga:
 - Un logo, un título y un subtítulo en su encabezado y enlaces a la Universidad de Jaén y al Congreso de los Diputados junto al copyright en el pie de página, siendo ambas partes constantes en todas las páginas.
 - Una estructura de páginas tal y como muestra la Ilustración 4.3.



Ilustración 4.3. Estructura propuesta para la aplicación web.

4.7. Justificación de la solución.

La solución propuesta sigue una serie de pasos incrementales característicos de la metodología ágil. Estos reflejan cómo desde unos documentos de texto se puede obtener una clasificación de emociones listas para presentarse en una aplicación web, por lo que se concluye que es la más conveniente a la hora de presentar el trabajo al cliente.

Se concreta más en el diseño de la interfaz, para la cual se proponen dos apartados principales: perfiles emocionales de un diputado a lo largo de su trayectoria en esta legislatura y los perfiles emocionales obtenidos en un día concreto, donde se muestren las emociones generales de esta sesión y las de cada diputado que haya comparecido.

Se considera que con estos dos apartados se pueden suplir las necesidades de los usuarios potenciales, los cuales pueden consultar información más general o más específica según su criterio.

4.8. Descripción de la solución.

En este apartado se incluye la información relativa a cada una de las iteraciones correspondientes al desarrollo del proyecto.

4.8.1. Iteración 1.

Iteración 1: “Como usuario quiero disponer de una base de trabajo con los diferentes documentos de texto del congreso de los diputados.”

Como se describió previamente en los apartados 2.1 y 2.2, los documentos están almacenados en el sitio web del congreso de los diputados con urls que siguen un patrón regular, por lo que en esta iteración el propósito será el de crear un crawler que nos descargue dichos documentos. Este crawler se encuentra en el archivo adjunto “TFGPrograma”, en un script denominado “crawler.py”.

Este crawler se ha diseñado específicamente para este proyecto y es bastante sencillo e intuitivo gracias a la naturaleza de las urls que guardan los documentos de las sesiones. Este incluye una función que actualizará los documentos almacenados en la carpeta “SOURCE/sesiones”, comprobando si el número de sesión que se intenta descargar está ya en dicho directorio y se ejecutará hasta que llegue al último que haya disponible en el sitio web de la cámara baja. El código también maneja las urls con excepciones, explicadas en el capítulo 2.2.

4.8.2. Iteración 2.

Iteración 2: “Como usuario quiero que el texto sobre el que se trabaja esté bien estructurado y almacenado para su posterior uso.”

Esta iteración hace referencia a lo descrito en el apartado 2.3, donde se explicaba la estructura de los documentos descargados y las referencias tomadas para que el resultado obtenido de los textos fuera lo más eficiente posible, esto es, ocupara la menor cantidad de memoria posible.

En el mismo directorio “TFGPrograma” se encuentra el script “readFiles.py” con el código debidamente comentado, explicando cómo son los pasos que se han seguido hasta obtener los diccionarios resultantes. Este script es el más extenso y contiene el código que más tarda en ejecutarse, ya que analiza los documentos uno por uno.

Contiene varias funciones:

- **readPDF:** Leerá los documentos, obteniendo solo el texto que nos interesa, es decir, desde que se abre o reanuda una sesión hasta que esta termina. También se guardará la fecha, clave del diccionario, y se localizarán los nombres de los diputados, claves de los subdiccionarios, con sus respectivas comparencias.
- **diccionarioDiputados:** función complementaria de la anterior, donde se crearán los subdiccionarios del tipo diputado – intervención, los cuales se asociarán a la clave principal, la fecha de sesión.
- Tres funciones de limpieza (`normalize_tokenize`, `stopper`, `stemmer`), las cuales eliminarán signos de puntuación, dividirán el texto por palabras, eliminarán las palabras vacías y obtendrán como resultado final los lemas.
- **preprocess:** última función que abre el diccionario base obtenido en “readPDF” y le aplica todas las funciones destinadas a la limpieza y depuración del texto, guardando también los diccionarios resultantes en un archivo local con extensión “`hpy`”.

4.8.3. Iteración 3.

Iteración 3: “Como usuario quiero que se extraigan las emociones de la mejor manera posible.”

Aquí se desarrollará lo comentado en el capítulo 3 de este documento, implementado en el script “`updateSources.py`” del mismo proyecto “TFGPrograma”.

4.8.3.1. Lectura de lexicón.

El script mencionado contiene una función denominada “readLexicon”, el cual abrirá el lexicón SEL (Díaz Rangel, Sidorov, & Suárez-Guerra, 2014). Este recurso tiene el formato mostrado en la Ilustración 4.4.

Palabra	Nula[%]	Baja[%]	Media[%]	Alta[%]	PFA	Categoría
abundancia	0	0	50	50	0.83	Alegría
acabalar	40	0	60	0	0.396	Alegría
acallar	50	40	10	0	0.198	Alegría
acatar	50	40	10	0	0.198	Alegría
acción	30	30	30	10	0.397	Alegría
aceptable	0	20	80	0	0.594	Alegría
aceptación	0	20	50	30	0.696	Alegría
acicate	10	50	40	0	0.429	Alegría
aclamación	10	10	10	70	0.799	Alegría
aclamar	10	10	10	70	0.799	Alegría
acogedor	0	0	50	50	0.83	Alegría
acoger	0	10	60	30	0.729	Alegría

Ilustración 4.4. Lexicón SEL.

El archivo contiene varios datos, pero los valores que nos interesan son la emoción que aparece en la columna “Categoría” y el valor de la columna “PFA” (Probability Factor of Affective use), que es un peso que indica en qué porcentaje esa palabra expresa la emoción asociada.

El resultado será un diccionario de tuplas con el formato: palabra – (id de una de las seis emociones, PFA). Una vez obtenido se guardará en un “npy” para el posterior análisis textual.

4.8.3.2. Clasificación de emociones.

Además, el script “updateSources.py” incluye la función más importante de la aplicación, la que obtendrá los resultados de los perfiles emocionales finales. Esta última se denomina “extractEmotions” y trabaja tanto con el diccionario de los lemas obtenidos en la segunda iteración como con el lexicón leído en la función previa. El funcionamiento de esta función se explica a continuación:

Se tendrá una lista con seis posiciones para cada uno de los perfiles analizados, es decir, para cada diputado, y cada una de estas posiciones equivaldrá a una emoción de Ekman. En un principio, todas estarán inicializadas a cero, es decir, serán contadores.

Entonces se recorre la lista de palabras asociadas a cada político y si una de ellas está en el lexicón leído previamente, se sumará su valor asociado, es decir, su peso emocional, al contador de su respectiva emoción. Sin embargo, existen dos excepciones:

- **Negación:** los negadores son palabras que pueden tener una carga negativa y, en consecuencia, pueden darle un significado distinto a los términos de una oración. Para tratar este problema, se ha decidido utilizar un lexicón de negadores y tras esto, analizar cada par de palabras, localizando aquellas formadas por un negador y la palabra que le sigue.

En un principio se puede presuponer que si una tupla no expresa alegría (por ejemplo, “no sonrío”) entonces expresará su emoción contraria: tristeza. Pero no tiene por qué, esta afirmación es bastante simplista y, por tanto, se ha optado por obviar estos términos negados.

- **Modificadores.** Existen una serie de palabras que amplifican o merman el significado de los términos que aparecen tras ellas. Un ejemplo de esto sería “bastante triste”, donde la emoción de tristeza se ve completamente ampliado por el adverbio “bastante”, y por tanto, su peso debe ser mayor a la hora de analizar un texto.

Para tratar esto se ha creado un lexicón con los modificadores más comunes, y se han dividido estos según intensifiquen o mermen en mayor o menor medida a las palabras que acompañan. El criterio que se ha seguido es el siguiente:

1. *algo, ligeramente, poco:* hacen que el peso de la palabra siguiente solo tenga valor de un 25% de su peso inicial.
2. *bastante, demasiado, muy:* aumentan el peso de la palabra siguiente un 75% de su peso inicial.
3. *completamente, sumamente:* aumentan el peso de la palabra siguiente un 75% de su peso inicial.

Una vez analizado cada par de palabras en busca de estas excepciones y obtenido el peso asociado a cada una de ellas (Ilustración 4.5), este valor se sumará al contador de la emoción que representa. El resultado para cada político es un vector con seis posiciones, donde cada una de ellas contiene un valor numérico que indica la presencia o la importancia que ha tenido la emoción correspondiente en ese texto.

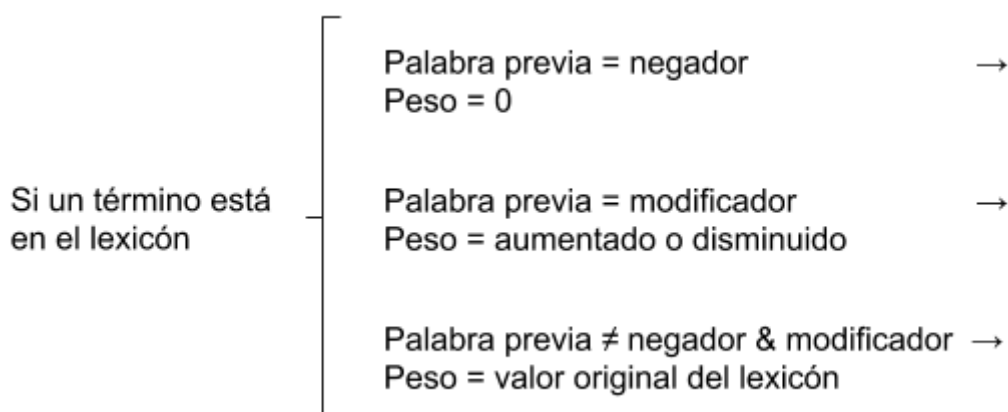


Ilustración 4.5. Esquema para la obtención del peso de las emociones.

Finalmente, para que el valor sea más representativo se opta por expresar los resultados en forma de porcentaje, es decir, se suman los seis valores para obtener un total y se divide cada uno de estos por ese total. Los resultados quedan guardados en un diccionario de diccionarios del tipo: fecha – diputado – lista de seis porcentajes de emociones.

4.8.3.3. Actualización de recursos y pruebas finales.

Por último, se incluye una función principal (main) que integra todo lo demás: descarga de pdfs, lectura de los mismos, procesamiento del texto y extracción de emociones. De este modo, solo ejecutando este script podremos obtener los datos resultantes de las emociones en forma de archivo “npy”, el cual se podrá incluir en la posterior interfaz.

A parte, existe un script denominado “extraPrintResults.py”, el cual ejecuta un breve programa que se mostrará por consola y que sirve para consultar algunos de los resultados.

4.8.4. Iteración 4.

Iteración 4: “Como usuario quiero disponer de una interfaz para poder interactuar con la aplicación creada de forma fácil e intuitiva.”

Esta última iteración incluye toda la fase de la creación de una interfaz web que permita a los usuarios interesados en el proyecto tener una percepción más sencilla de los resultados.

Para implementarla se ha hecho uso de [Flask](#)³, un complemento de Python que permite trabajar en este lenguaje a la hora de crear una interfaz web. De este modo se podrá usar el diccionario obtenido en la iteración anterior para obtener las diferentes medidas que se irán mostrando en las distintas páginas de nuestra web.

Todo este trabajo está situado en el directorio del proyecto adjunto “TFGWeb”, el cual incluye un diseño para cada una de las páginas propuestas anteriormente (Ilustración 4.3). Este incluye dos scripts de Python:

- `app.py`: archivo principal y ejecutable, el cual pondrá en marcha la aplicación web.
- `globalvariables.py`: usado para obtener los diferentes diccionarios dependiendo de la información relativa a las emociones que se quiera mostrar. Aquí se abren los diccionarios guardados previamente (localizados en la carpeta “data”) en formato “.npy” para hacer uso de ellos desde la aplicación principal. No se ha utilizado una base de datos puesto que es un contenido estático, con el que los usuarios no deben interactuar, solo han de verlo.

Además, en el directorio “templates” se encuentran las plantillas de las distintas páginas en formato html y en “static” todos los demás recursos, es decir, los archivos CSS, JavaScript y las imágenes utilizadas.

³ <http://flask.pocoo.org/>

4.9. Herramientas para la implementación del sistema.

En este apartado se exponen las diferentes tecnologías utilizadas en el desarrollo de la aplicación del proyecto.

4.9.1. Python.

Python (Ilustración 4.6) es el lenguaje de programación utilizado para el Procesamiento del Lenguaje Natural, y es el que se ha usado para extraer y analizar las emociones a partir del texto de las diferentes declaraciones hechas por los diputados, en concreto se ha utilizado Python3.



Ilustración 4.6. Logo de Python.

Este lenguaje de programación se caracteriza por la sencillez y la efectividad de sus comandos, haciendo fácil la legibilidad del código. Además, soporta la programación orientada a objetos y permite organizar bien el código implementado.

Además de lo anterior, se ha escogido este lenguaje de programación por lo siguiente:

- Python provee al programador de una gran cantidad de librerías casi para todo tipo de usos, por ejemplo para leer PDFs o para guardar diccionarios en un archivo, como se ha hecho en este caso.
- El código se crea a partir de indentaciones, es decir, no hace falta utilizar corchetes o tantos paréntesis para que el código funcione, por lo que hace que el resultado final se mucho más legible.
- Lenguaje interpretado, sin necesidad de compilar el código antes de ejecutarlo, e interactivo, lo cual nos permite realizar operaciones desde la línea de comandos y producir resultados visibles por medio de esta.

4.9.2. Flask.

Flask (Ilustración 4.7) es un microframework de Python, el cual permite crear aplicaciones web de forma sencilla. Esto es así porque toda la funcionalidad de la web puede implementarse en Python, que como se ha dicho anteriormente, es un lenguaje fácil de utilizar.



Ilustración 4.7. Logo de Flask.

Puesto que la aplicación base está hecha en Python, añadiendo esta extensión se puede implementar la interfaz web a partir de esta sin necesidad de convertirla a otro lenguaje, ya que esta no incorporará elementos complejos.

4.9.3. HTML, CSS, JavaScript y JQuery.

HTML, CSS y JavaScript son los lenguajes dirigidos a la programación web, siendo HTML un lenguaje de etiquetas para la web, CSS es usado para dar estilo a las páginas implementadas y finalmente JavaScript añadirá la funcionalidad. Estos tres elementos unidos a Flask darán forma a la interfaz del proyecto.



Ilustración 4.8. Logos de HTML, CSS, JavaScript y JQuery.

Además, se ha incluido JQuery, una librería específica realizada en JavaScript, la cual permite añadir diferentes funcionalidades, como el menú desplegable o el calendario.

4.9.4. Google Charts



Ilustración 4.9. Logo de Google Charts.

Por último, para la funcionalidad de las gráficas añadidas a la interfaz web se ha utilizado [Google Charts](https://developers.google.com/chart/)⁴, uno de los muchos complementos que incluye Google para construir nuestro propio sitio web. Permite crear tablas incluyendo un código sencillo, el cual solo hay que modificar para adaptarlo a nuestros datos concretos.

⁴ <https://developers.google.com/chart/>

5. CONCLUSIONES Y TRABAJOS FUTUROS.

En este capítulo se explican las conclusiones obtenidos tras la realización de este trabajo además de las posibles mejoras que pudieran incluirse en un futuro.

5.1. Conclusiones.

Decidí aceptar y por tanto, embarcarme en la realización de este Proyecto de Fin de Grado, debido al interés y curiosidad que me genera el ámbito del Procesamiento el Lenguaje Natural (PLN). Una vez estudiada esta asignatura, quise saber más.

A la parte que se dedica a la extracción de emociones basándose en textos le queda mucho por avanzar a pesar de las numerosas aplicaciones que puede tener, como la descrita en este trabajo, más aún si se trabaja con textos en castellano. Por eso mismo encontré el tema interesante y si le sumamos el seguimiento que suelo hacer del día a día político encontré el trabajo ideal.

Gracias a lo desarrollado he podido aprender más sobre las emociones y el tratamiento de los textos, sobre lo tedioso que puede llegar a ser trabajar con lexicones y adaptarlos a un problema concreto y lo difícil que es extraer resultados fiables respecto a la emociones. Sin embargo, tras todo este trabajo, el hecho de ver los datos obtenidos merece la pena.

Además, destaco la parte de la creación de la aplicación web. Nunca había utilizado “flask” y mis conocimientos sobre programación web (html, css, js) eran prácticamente nulos, por lo que he podido aprender mucho gracias a este proyecto.

5.2. Trabajos futuros.

Como ya se ha explicado a lo largo de este trabajo, aún queda un largo camino que recorrer para que finalmente el análisis que se realiza de las emociones sea lo más realista posible. Además también hay que hacer hincapié en los métodos de Procesamiento del Lenguaje Natural para que estos se adapten mejor al castellano y sus particularidades como idioma.

Aparte de lo comentado previamente, la aplicación aquí expuesta podría tener diversas mejoras como las siguientes:

- Si la aplicación se instalara en un servidor permanente la web podría ser dinámica, haciendo uso de una base de datos que se actualizara cada x tiempo (por ejemplo, una vez a la semana) y de esta manera, si hubiera nuevos documentos de sesiones se podrían analizar para añadir estos nuevos datos a la colección.
- Se podrían obtener más gráficas y resultados basándonos en los datos que ya tenemos. Por ejemplo, sería interesante incluir una nube de palabras por emoción para saber cuáles son las causantes de cada emoción, ya sea para una sesión en general o un diputado concreto.
- Añadir un apartado de temáticas. Se podrían clasificar los documentos por los temas tratados y ver qué emociones se generan dependiendo de lo que se esté hablando. Existe un resumen de los temas a tratar en cada pdf por lo que sería factible.
- Etiquetar manualmente cada pdf con el propósito de obtener un lexicón más específico al tema que se está tratando, en vez de usar uno genérico.

Bibliografía

- Canales, L., & Martínez-Barco, P. (2014). Emotion Detection from text: A Survey. *University of Alicante*.
- Díaz Rangel, I., Sidorov, G., & Suárez-Guerra, S. (2014). Creación y evaluación de un diccionario marcado con emociones y ponderado para el español. *Onomazein* 29.
- Ekman, P., Friesen, W. V., & Ellsworth, P. (1972). *Emotion in the human face: Guidelines for research and integration of findings*. New York: Permagon.
- Lövheim, H. (2012). A new three-dimensional model for emotions and monoamine neurotransmitters. *Medical Hypotheses Volume 72, Issue 2*, 341-348.
- Plutchik, R., & Kellerman, H. (1986). *Emotion: Theory, Research and Experience Vol. 3*. New York: Academic Press.
- Shaver, P., Schwartz, J., Kirson, D., & O'connor, C. (1987). Emotion knowledge: Further exploration of a prototype approach. *J. Pers. Soc. Psychol.* 52,6, 1061.
- Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., . . . Gordon, J. (2012). Empirical Study of Opinion Mining in Spanish Tweets. *LNAI 7630*, 1-14.
- Stone, P., Dunphy, D. C., Smith, M. S., & Ogilvie, D. M. (1968). The general inquirer: A computer approach to content analysis. *J.Region. Sci.* 8, 1, 113-116.
- Strapparava, C., & Valiutti, A. (2004). WordNet-Affect: an affective extension of wordnet. *Proceedings of the 4th International Conference on Language Resources and Evaluation*, (págs. 1083-1086).
- Yadollahi, A., Gholipour Shahraki, A., & R. Zaiane, O. R. (2017). Current State of Text Sentiment Analysis from Opinion to Emotion Mining. *ACM Computing Surveys, Vol. 50, No.2, Article 25*.

ANEXO A. Manual de instalación.

A continuación se explicará el proceso de instalación de los componentes que harán posible que nuestro sistema funcione. Para ello habrá que seguir unos sencillos pasos.

A.1. Instalación de Python 3.

El proyecto se ha desarrollado en Python 3 y se ha hecho uso de sus librerías para cada una de las necesidades planteadas, por lo tanto es imprescindible instalarlo para poder ejecutar el programa.

Para ello entramos a la [web oficial de Python](https://www.python.org/downloads/)⁵ y buscamos la última versión de Python 3 para el sistema operativo que se esté utilizando, ya que la aplicación web funcionará independientemente de cuál usemos.

A.2. Instalación de pip.

Pip es un paquete que permite instalar librerías y componentes de Python de forma sencilla. Para instalarlo hay que seguir los siguientes pasos:

Si utilizamos Ubuntu basta con ejecutar los siguientes comandos en consola:

```
sudo apt-get install python3-setuptools  
sudo easy_install3
```

Si por el contrario se hace uso de Windows, el complemento vendrá en un ejecutable por defecto y se podrá utilizar sin necesidad de configurar nada.

A.3. Instalación de paquetes de Python con pip.

El uso de pip es sencillo, solo hace falta un comando para instalar los paquetes necesarios, tanto en Windows como en Ubuntu:

```
pip3 install [librería]
```

Se sustituirá “librería” con cada uno de los paquetes necesarios para la aplicación, que son los siguientes:

- Flask: Framework que permite crear aplicaciones web.
- numpy: Librería de funciones matemáticas.
- PyPDF2: Librería para manejar pdfs.

⁵ <https://www.python.org/downloads/>

A.4. Ejecución del proyecto.

Ahora que todas las librerías y paquetes están instalados ya se puede ejecutar la aplicación web. Para ello abrimos la consola de comandos y nos vamos al directorio donde está situada la web del proyecto, el cual dependerá de donde se haya almacenado este, y ejecutamos el script principal, esto es, “app.py”. El comando sería:

```
python app.py
```

```
python3 app.py si tenemos otra versión de Python instalada también.
```

En la misma consola aparecerá la dirección donde será visible la web, bastará con clicar en ella o copiarla en algún navegador para poder ver el resultado.

En la Ilustración A.1 se puede ver un ejemplo de este proceso en una consola de Windows. Primero nos dirigimos al directorio donde se encuentra la web, en este caso “PycharmProjects/TFGWeb” y después ejecutamos “app.py”. La url de la web será “127.0.0.1/5000”.

```
C:\Users\Andrea>cd PycharmProjects/TFGWeb  
  
C:\Users\Andrea\PycharmProjects\TFGWeb>python app.py  
* Serving Flask app "app" (lazy loading)  
* Environment: production  
  WARNING: Do not use the development server in a production environment.  
  Use a production WSGI server instead.  
* Debug mode: on  
* Restarting with stat  
* Debugger is active!  
* Debugger PIN: 208-416-217  
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Ilustración 0.1. Puesta en marcha de la aplicación web.

ANEXO B. Manual de usuario.

El objetivo de este anexo es el de elaborar una guía para los usuarios que quieran utilizar el sitio web.

Lo primero que aparecerá al dirigirlos a la url de la aplicación web será la página de inicio, que contendrá tanto el encabezado como el pie de página que serán comunes a todas las páginas del sitio web. En el encabezado encontramos el título y el subtítulo además del logo, mientras que en el pie de página encontramos tanto el logo de la Universidad de Jaén como el del Congreso de los Diputados, además de los derechos de autor.

Desde aquí tendremos dos opciones: buscar información por sesiones concretas o la relativa a un diputado. Todo lo descrito se puede ver en la Ilustración B.1 y en la Ilustración B.2.



Ilustración 0.1. Inicio de página: opción "sesión completa".



Ilustración 0.2. Inicio de página: opción "diputados".

B.1. Búsqueda por sesión.

Si vamos a la opción de “sesión completa”, nos aparecerá una página tal y como se muestra en la Ilustración B.3.



EMOCIONES EN EL CONGRESO
¿Cómo se sienten nuestros diputados?

Sesiones Disponibles

« Julio 2018 »						
L	M	X	J	V	S	D
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

¿Qué se habló ese día?

Selecciona diputado:

Perfil del diputado seleccionado →

PERFIL COMPLETO DE SESIÓN →

UJA Congreso de los Diputados

© Andrea López Fernández

Ilustración 0.3. Página de selección de sesión.

Nos aparecerá un calendario donde podremos seleccionar solamente las fechas en las que se ha celebrado un pleno en el congreso de los diputados, las cuales aparecen marcadas en color verde. Una vez hecho esto, se podrá consultar en una pestaña nueva el PDF relativo a ese día en el enlace “¿Qué se habló ese día?”

Una vez seleccionada la fecha tenemos dos opciones:

- Ver el perfil completo de la sesión, donde se podrá ver un estudio generalizado de las emociones detectadas ese día.
- Ver el perfil de un diputado concreto. Aparecerá una lista desplegable con los diputados que han comparecido ese día y se podrá ver su estado emocional en ese día concreto.

Si se intenta clicar sobre alguna de estas opciones sin haber seleccionado la fecha o el diputado en cada caso, aparecerá un mensaje que avisará del error. Un ejemplo de esto se puede observar en la Ilustración B.4, donde el error nos avisa de que no hemos seleccionado una fecha de la cual mostrar la información.

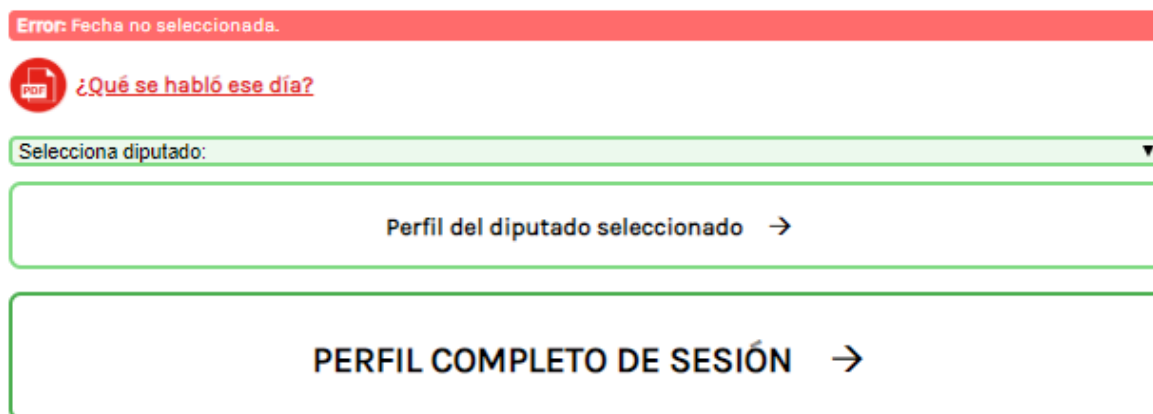
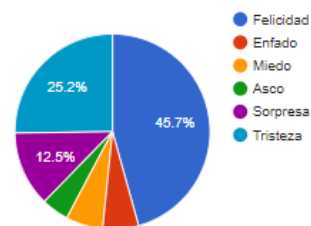


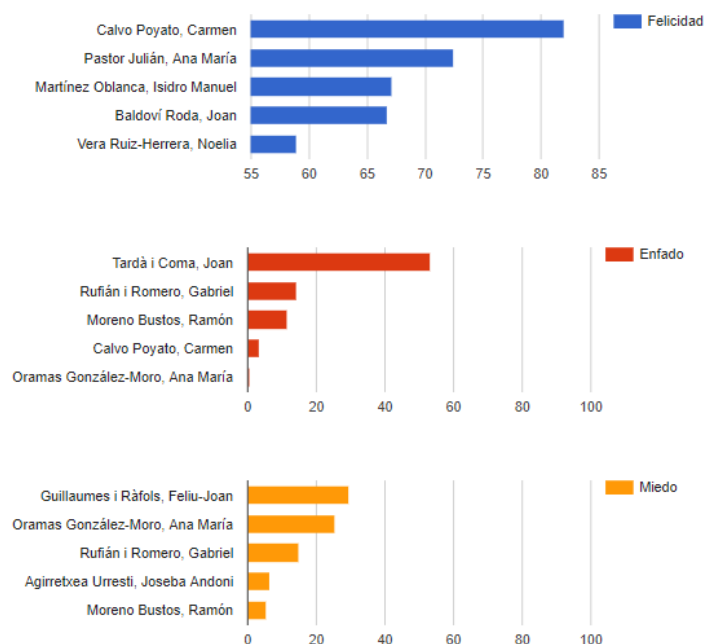
Ilustración 0.4. Ejemplo de error si no se selecciona una fecha.

B.1.1. Emociones en una sesión concreta.

Si vamos al perfil general de una sesión en un día concreto nos encontraremos con dos apartados. El primero será un diagrama de sectores, que nos indicará el porcentaje en el que ha estado presente cada emoción ese día. Un ejemplo de esto se puede ver en la Ilustración B.5.

Sesión 4/7/2018**Perfil emocional general:****Ilustración 0.5. Página de sesión: diagrama de sectores.**

El segundo apartado (Ilustración B.6.) incluirá una serie de rankings, uno por cada emoción analizada, donde se mostrarán hasta los cinco diputados que más hayan sentido esas emociones.

Ranking por emociones:**Ilustración 0.6. Ranking por emociones en una sesión.**

B.1.2. Emociones de un diputado en una sesión concreta.

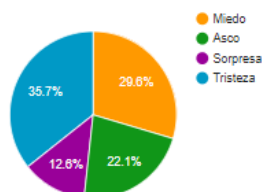
Al igual que en el apartado anterior, lo primero que aparecerá también será un diagrama de sectores que representa los porcentajes de cada emoción.

Sin embargo esta vez, en vez de un ranking, se muestra un gráfico de barras donde se puede comparar entre las emociones que se han analizado este día concreto y las emociones que suele expresar el diputado seleccionado (su perfil habitual).

Todo lo descrito en este apartado se ve reflejado en la Ilustración B.7.

Sesión 4/7/2018: Guillaumes i Ràfols, Feliu-Joan

Perfil emocional del diputado en esta sesión:



Comparación con el perfil habitual del diputado:

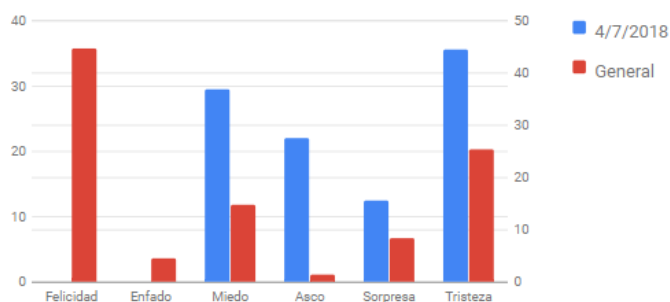


Ilustración 0.7. Emociones de un diputado en una sesión concreta.

B.2. Búsqueda por diputado.

Desde la página de inicio tenemos otra opción: “Diputados” (Ilustración B.2). Si no nos interesa algo tan específico como una sesión de un día particular, sino las emociones que muestra un político en general, esta es la opción correspondiente.

En la Ilustración B.8 podemos ver la página que se nos abre, donde aparece una lista de listas con los diferentes grupos parlamentarios y sus respectivos diputados escritos en orden alfabético (Ilustración B.9). Si han hecho alguna intervención, entonces es posible seleccionarlos para ver sus estadísticas.

Lista de Diputados



Ilustración 0.8. Lista de diputados.



Ilustración 0.9. Lista de diputados de un grupo parlamentario seleccionado.

B.2.1. Diputado concreto.

En esta última página (Ilustración B.10) es posible consultar el perfil emocional habitual del diputado seleccionado, expresado mediante un gráfico de sectores y, además, un histórico en el que se va viendo la evolución de este diputado según el día en el que haya intervenido. Este último diagrama se puede ampliar y reducir y permite seleccionar un día concreto, pudiendo así acceder a la página del apartado B.1.1., la correspondiente a la sesión de ese día.

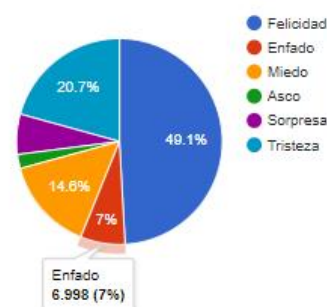
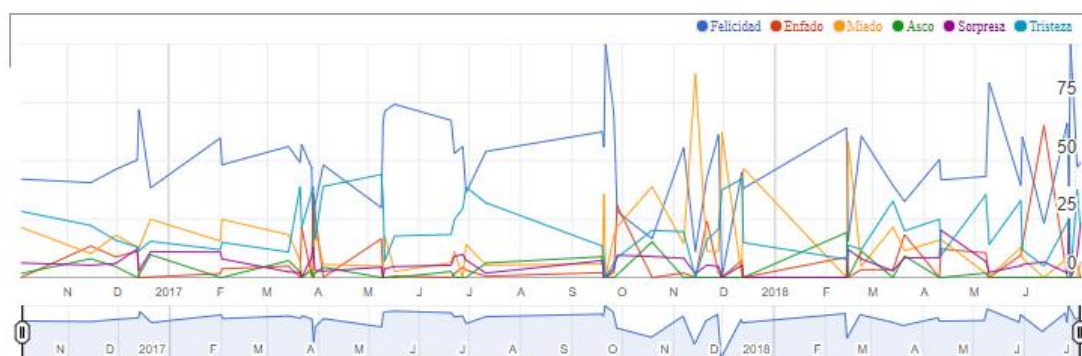
Agirretxea Urresti, Joseba Andoni**Perfil emocional del diputado:****Evolución del diputado a lo largo de las sesiones:**

Ilustración 0.10. Emociones habituales de un diputado.

ANEXO C. Índice de Ilustraciones.

Ilustración 1.1. Diagrama de Gantt.	11
Ilustración 2.1. Web del congreso de los diputados.	15
Ilustración 2.2. Ejemplo de url de pdf.	16
Ilustración 2.3. Extracto de pdf de sesión.	17
Ilustración 2.4. Estructura de un diccionario.	18
Ilustración 3.1. Definición de "emoción" en la RAE.	20
Ilustración 3.2. Emociones de Ekman.	22
Ilustración 4.1. Etapas en cada iteración de la metodología ágil.	28
Ilustración 4.2. Metodología Kanban.	28
Ilustración 4.3. Estructura propuesta para la aplicación web.	31
Ilustración 4.4. Lexicón SEL.	34
Ilustración 4.5. Esquema para la obtención del peso de las emociones.	36
Ilustración 4.6. Logo de Python.	38
Ilustración 4.7. Logo de Flask.	39
Ilustración 4.8. Logos de HTML, CSS, JavaScript y JQuery.	39
Ilustración 4.9. Logo de Google Charts.	40
Ilustración A.1. Puesta en marcha de la aplicación web.	45
Ilustración B.1. Inicio de página: opción "sesión completa".	46
Ilustración B.2. Inicio de página: opción "diputados".	46
Ilustración B.3. Página de selección de sesión.	47
Ilustración B.5. Ejemplo de error si no se selecciona una fecha.	48
Ilustración B.6. Página de sesión: diagrama de sectores.	49
Ilustración B.7. Ranking por emociones en una sesión.	49
Ilustración B.8. Emociones de un diputado en una sesión concreta.	50
Ilustración B.9. Lista de diputados.	51
Ilustración B.10. Lista de diputados de un grupo parlamentario seleccionado.	51
Ilustración B.11. Emociones habituales de un diputado.	52

ANEXO D. Índice de Tablas.

Tabla 1.1. Planificación temporal del proyecto.	11
Tabla 1.2. Coste de personal.	13
Tabla 1.3. Coste total del proyecto.	13
Tabla 3.1. Modelos de emociones.	22
Tabla 3.2. Número de palabras del lexicón SEL.	24
Tabla 4.1. Historias de usuario.	29