

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

TRABAJO TERMINAL

**Recolector y clasificador de
noticias**

2018-B013

PRESENTAN:

CARLOS ANDRES HERNANDEZ GOMEZ
LUIS DANIEL MEZA MARTÍNEZ

DIRECTORES:

Dr. JOEL OMAR JUÁREZ GAMBINO
Dra. CONSUELO VARINIA GARCÍA
MENDOZA



Ciudad de México, 27 de noviembre de 2019

Índice general

1. Introducción	1
1.1. Problemática	2
1.2. Justificación	2
1.3. Solución Propuesta	3
1.4. Objetivo general	3
1.5. Objetivos Específicos	3
2. Estado del arte	5
2.1. Introducción	5
2.2. Trabajos nacionales	6
2.2.1. Clasificación de noticias de diarios	6
2.2.2. Desastres naturales en México	6
2.2.3. Extraer información de noticias de DN	7
2.3. Trabajos I	7
2.3.1. Clasificador de noticias usando autoencoders	7
2.3.2. Document classification for newspaper articles	8
2.3.3. Category classification and topic discovery	8
2.3.4. Automatic news articles classification in indonesian	10
2.3.5. News article text classification in indonesian language	11
2.4. Herramientas D	12
2.4.1. Cloud natural language	12
2.4.2. Googlebot	12
2.4.3. Watson natural language classifier	12
3. Marco teórico	14
3.1. Inteligencia Artificial	15
3.2. Procesamiento de LN	16
3.2.1. Pre-procesamiento	16

3.3. Aprendizaje Automático	19
3.4. AA Para texto	20
3.4.1. Representación del T	21
3.5. A Supervisado	23
3.6. Clasificación	23
3.6.1. Regresión logística	23
3.6.2. Naive bayes	24
3.6.3. Máquina de soporte vectorial	26
3.6.4. Random forest	26
3.7. Métricas de evaluación	27
3.8. Validación cruzada	29
3.9. Web Scraping	30
3.9.1. Técnicas de web scraping	30
3.10. Crawler	31
3.11. Python	31
3.11.1. Scrapy	32
3.12. Aplicación Web	33
3.13. Patrón de diseño	33
3.14. Modelo Vista Controlador	33
3.15. Framework	34
3.16. JavaServer Faces	34
3.17. Internet	34
3.17.1. World Wide Web	35
3.18. HTML	35
3.19. Sitios Web	36
3.19.1. Página Web	36
3.19.2. Blog	36
3.19.3. Foro	37
4. Análisis y diseño	38
4.1. Actores y roles	38
4.2. Requisitos f.	38
4.3. Requisitos no f.	39
4.4. Reglas de negocio	40
4.5. Casos de uso	43
4.5.1. Diagrama de casos de uso	43
4.5.2. CU1 Recolectar noticias	44
4.5.3. CU2 Clasificar noticias	47

4.5.4.	CU3 Pre-procesar noticias	48
4.5.5.	CU4 Mostrar resultados	50
4.6.	Mensajes	53
4.7.	Pantallas	54
4.7.1.	UI1 Página de inicio	54
4.7.2.	UI2 Recolección y clasificación	55
4.7.3.	UI3 Proceso concluido	56
4.7.4.	UI4 Resultados de consulta	57
4.8.	Diagrama de secuencia	60
5.	Desarrollo	61
5.1.	Recolección	62
5.1.1.	Selección de sitios web	62
5.1.2.	Análisis de sitios web	66
5.1.3.	Creación de recolector	66
5.1.4.	Recolección de noticias	68
5.2.	Entrenar C. ■	73
5.2.1.	Preprocesamiento	74
5.2.2.	Entrenamiento	82
5.2.3.	Selección	92
5.2.4.	Pruebas	93
5.2.5.	Persistencia	95
5.3.	Aplicación	96
5.3.1.	Selección	97
5.3.2.	Recolectar	98
5.3.3.	Clasificar	99
5.3.4.	Mostrar resultados	101
6.	Conclusión ←	104
6.1.	Conclusiones	104
6.2.	T. Futuro ■	106
	Bibliografía	107

Esta sección se deba
llamar "Conclusiones y
trabajo futuro"

Índice de figuras

3.1. Campo de estudio	14
3.2. Matriz de confusión	27
3.3. Validación cruzada	29
3.4. Etapas del proceso de <i>Web scraping</i>	31
3.5. Patrón MVC Modelo Vista Controlador.	34
4.1. Diagrama de casos de uso	43
4.2. Pantalla UI1 Inicio	55
4.3. Pantalla UI2 Espera de proceso	56
4.4. Pantalla UI3 Proceso concluido	57
4.5. Pantalla UI4 Resultados de consulta	59
4.6. Pantalla UI5 Cambio de periodo	59
4.7. Diagrama de secuencia	60
5.1. Etapas de desarrollo	61
5.2. Etapas de la recolección	62
5.3. Nivel de confianza de usuarios al consultar un sitio web.	63
5.4. Ranking de sitios de noticias del período de enero del 2018 a enero del 2019.	63
5.5. Visitantes únicos a sitios de noticias durante el año 2018.	64
5.6. Proceso de recolección	67
5.7. Noticias recolectadas durante el primer corte.	69
5.8. Noticias recolectadas al finalizar el segundo corte.	70
5.9. Noticias recolectadas por sitio web al finalizar el segundo corte	70
5.10. Noticias recolectadas secciones	71
5.11. Diferencias del clasificador en el TT 2017-A042 y 2018-B013	73
5.12. Proceso de entrenamiento	74
5.13. Etapas de preprocesamiento	75

5.14. Noticias por sección	77
5.15. Noticias por sección	78
5.16. Corpus de entrenamiento	81
5.17. Corpus de prueba	82
5.18. Etapas de entrenamiento	83
5.19. Corpus de entrenamiento	87
5.20. Kernel de la función (Pedregosa et al., 2011)	90
5.21. Kernel radial (Pedregosa et al., 2011)	90
5.22. Etapas de la aplicación web	96
5.23. Pantalla de Inicio	97
5.24. Mensaje de espera	98
5.25. Mensaje de error en la recolección	99
5.26. Proceso de clasificación	100
5.27. Mensaje que se muestra una vez clasificadas las noticias	101
5.28. Funcionalidad de la aplicación	102
5.29. Vista de las noticias recolectadas	103
5.30. Vista de las noticias recolectadas del día de ayer	103

Índice de cuadros

5.1. Secciones de los sitios web	65
5.2. Ejemplo de estructura de un archivo CSV	68
5.3. Noticias recolectadas por sitio web	72
5.4. Identificador de sitio web	80
5.5. Etiquetas de secciones	86
5.6. Biblioteca de algoritmo	86
5.7. Naive bayes	91
5.8. Regresión logística	91
5.9. Random Forest	91
5.10. Máquina de soporte vectorial	92
5.11. Precisión de los mejores parámetros	93
5.12. Número de noticias	93
5.13. Matriz de confusión	94
5.14. Métricas de evaluación	94

Capítulo 1

Introducción



El artículo periodístico o noticia, es la información de un hecho de interés ocurrido en un periodo de tiempo determinado. Constituye el elemento primordial en la información de la prensa y del género básico del periodismo ([Internaútica, 2018](#)). Conocer los acontecimientos del mundo independientemente del tema, día o lugar en el cual se han suscitado, tiene una gran importancia en la sociedad, se comparten por distintos medios de comunicación, tales como la televisión, redes sociales, diarios, blogs y la radio. Nos permiten conocer la situación económica del país, logros de la ciencia, desastres naturales, la situación en cuestión de inseguridad entre otros hechos. En el ámbito de las inversiones, crean expectativas y eso a su vez puede modificar los planes de inversión en cualquier sector, siendo así de suma importancia compartirlas de una forma eficaz ([Manning et al., 2010](#)).

El uso de páginas web como medio de comunicación está en incremento, permitiendo consultar noticias de distintos sitios como los periódicos electrónicos; su información al igual que un diario tradicional se encuentra dividida

en secciones para facilitar la consulta, sin embargo, la clasificación suele variar en cada portal, incluso teniendo el mismo contenido. Un problema mayor se encuentra en los sitios independientes, los cuales no cuentan con una segmentación particular, haciendo difícil realizar una búsqueda eficaz.

1.1. Problemática

Los métodos tradicionales para la recopilación de información de los recolectores web (*Crawler*), están basados en las etiquetas o marcadores que los sitios añaden a su código fuente, por ejemplo, algunos artículos periodísticos son etiquetados a la sección que pertenecen (política, deporte, cultura, etc). Sin embargo, existen muchas fuentes de información que no etiquetan sus publicaciones, incluso si la tarea es realizada, dicha segmentación no indica claramente el tipo de contenido; al consultar algunos de los portales mas visitados en México (en el giro del periodismo) se encuentra definida la sección deportes con varios sinónimos como **Universal deportes** (diario El Universal), **La afición** (Milenio), **Adrenalina** (Excelsior), etc. Como este ejemplo se encuentran más. Las noticias son segmentadas de forma tan diversa que ha complicado su búsqueda en la Internet.

Para definir las etiquetas o marcadores con los cuales se clasifica la información de los sitios web, se requiere un proceso manual de análisis de la información. Este proceso implica tiempo y esfuerzo por parte de las personas que realizan el trabajo. Por lo anterior se plantea la necesidad de crear métodos para automatizar esta tarea.

1.2. Justificación

Hoy en día existen distintas maneras de informarse acerca de los acontecimientos más recientes, por ejemplo, la televisión, blogs, redes sociales, foros, diarios, etc. Esto ha provocado que la información se encuentre dispersa y se deba acceder a múltiples recursos para ser recopilada, implicando una inversión de tiempo y esfuerzo. Para facilitar esta tarea, existen herramientas que hacen la búsqueda de noticias de interés para el usuario en forma automática. Sin embargo, dichas herramientas requieren que los sitios a consultar tengan

etiquetas definidas y homogéneas.

Según el diario El Economista ([Economista, 2019](#)) el sitio web Animal Político¹ ocupa el lugar número cuatro en el ranking de medios nativos digitales, clasifica sus noticias de una manera poco habitual para los lectores como la sección **El sabueso**, **El plumaje**, **Hablemos de . . .**, entre otras, lo que hace complicado obtener los artículos con los métodos tradicionales de recopilación que, se basan sólo en las etiquetas que identifican cada sección y no el contenido de las noticias.

1.3. Solución Propuesta

Se propone crear una aplicación web que recolecte y clasifique noticias de acuerdo a su contenido y periodo de publicación. Finalmente, las noticias que satisfagan ambos filtros (Tipo de contenido y fecha de publicación) serán mostradas al usuario.

1.4. Objetivo general

Crear un recolector de noticias, el cual permita recopilar información de diferentes fuentes como diarios, sitios de noticias, foros y mediante el análisis automático de su contenido muestre aquellas noticias que satisfagan los filtros establecidos por el usuario.

1.5. Objetivos Específicos

- Desarrollar un recolector de noticias, el cual permita obtener información de diferentes fuentes como diarios, sitios de noticias, blogs y foros
- Analizar de forma automática el contenido de las noticias para satisfacer los filtros establecidos por el usuario
- Mostrar las noticias que cumplieron con los filtros establecidos, así como su enlace (URL) para redirigirlos a la página de la noticia

¹www.animalpolitico.com

- Afinar el clasificador de noticias realizado en el trabajo terminal 2017-A042 para utilizarlo en el contexto de esta propuesta (filtro de sección)

Capítulo 2

Estado del arte



2.1. Introducción

El uso de la información digital ha superado la producción de libros y publicaciones impresas, este fenómeno ha influenciado la producción de bibliotecas digitales, publicaciones electrónicas; se ha incrementado el uso de las redes sociales, correos electrónicos, creando un gran repositorio de información útil, el cual puede ser analizado([Aggarwal, 2018](#)).

Debido a la necesidad de procesar grandes volúmenes de datos recolectados de Internet, se han desarrollado diversas investigaciones entorno a esta tarea. A continuación se muestran distintos artículos nacionales e internacionales relacionados al campo de investigación (clasificación de noticias), de igual forma se muestran herramientas web que desempeñan un trabajo similar

al propuesto (sitio web de noticias). Cabe destacar que el área de interés cuenta con un amplio desarrollo, no obstante solo se mencionan los trabajos más relevantes para este documento.

2.2. Trabajos nacionales

2.2.1. Clasificación de noticias de diarios de circulación nacional mediante aprendizaje automático

En este trabajo terminal de la Escuela Superior de Cómputo ([García et al., 2018](#)) los autores clasifican mediante técnicas de aprendizaje automático, noticias de diarios de circulación nacional en las diferentes secciones en que en estos se dividen. Se recolectaron 4,027 artículos de tres diarios de circulación nacional: **El universal**, **La jornada** y **Excélsior**. 3,624 noticias fueron utilizadas para la etapa de entrenamiento y 407 para hacer las pruebas.

El trabajo utiliza pre-procesamiento de información con la técnica tokenización y lematización (ver [Capítulo 3](#)). El mejor resultado en las pruebas se dio en la combinación del algoritmo **TF-IDF** para extraer las características y **Máquinas de soporte vectorial** para la clasificación de artículos, se obtuvo un 79.81 % de exactitud, *i.e* 8 de cada 10 noticias son clasificadas correctamente.

2.2.2. Clasificación automática de textos de desastres naturales en México

En este trabajo se propone clasificar noticias en el ámbito **desastres naturales** ([Téllez-Valero et al., 2019](#)), utilizando estrategias de reducción de dimensionalidad conocidas como, umbral en la frecuencia y ganancia en la información, los métodos de clasificación utilizados fueron el clasificador simple de Bayes y vecinos más cercanos.

Se utilizaron 375 noticias del periódico Reforma como conjunto de entrenamiento, se clasificaron en artículos relevantes e irrelevantes, de los cuales 11.5 % de noticias eran relevantes y el 88.5 % restante eran irrelevantes. Una vez obtenido el conjunto de noticias se procedió con un pre-procesamiento,

el cual reduce el tamaño de los documentos, eliminando la parte de los textos que no brindan información útil, posteriormente se realizó un indexado: Los documentos son representados por vectores de palabras en un espacio de dimensión n , para realizar una reducción de dimensionalidad. Finalmente se utilizaron técnicas de clasificación (Algoritmo simple de Bayes) con el cual se obtuvo un resultado de 97 % de efectividad en la clasificación de noticias.

2.2.3. Usando aprendizaje automático para extraer información de noticias de desastres naturales

Este trabajo describe un sistema basado en métodos de Aprendizaje automático que mejora la adquisición de datos de desastres naturales (Téllez Valero et al., 2009). Este sistema automáticamente llena una base de datos de desastres naturales con la información extraída de noticias de periódicos en línea. En particular, se extrae información acerca de cinco tipos de desastres naturales: huracanes, temblores, incendios forestales, inundaciones y sequías. Los algoritmos implementados para la extracción de información son los siguientes:

- Naive bayes
- Maquinas de soporte vectorial
- C4.5

Los resultados experimentales en una colección de noticias en Español muestran la eficacia del sistema propuesto tanto para detectar documentos relevantes sobre desastres naturales (alcanzando una medida-F de 98 %), así como para extraer hechos relevantes para ser insertados en una base de datos dada (alcanzando una medida-F de 76 %).

2.3. Trabajos internacionales

2.3.1. Clasificador de noticias usando autoencoders

En este trabajo se propone la clasificación de noticias utilizando *Deep Learning* (Farias et al., 2018), las noticias se clasificaron en las siguientes categorías:

- Deportes

- Política
- Espectáculos
- Economía
- Policía

El alcance que tiene es:

- Local (Valparaíso)
- Nacional (Chile)
- Internacional (resto del mundo)

El clasificador se construyó utilizando una base de datos con 542 noticias etiquetadas con los criterios anteriores, las características se obtuvieron utilizando Autoencoders (AE) para entrenar una Red Neuronal Artificial (ANN). Los resultados obtenidos con 156 noticias fue una tasa de éxito del 92.3 % para la clasificación de la categoría y un 87.2 % para el clasificador de alcance. La tasa general de éxito, categoría y alcance fue de 83.75 %.

2.3.2. Document classification for newspaper articles

El trabajo clasifica artículos de la universidad *Massachusetts Institute of Technology* ([Ramdass and Seshasai, 2009](#)) en las categorías: *Arts, Features, News, Opinion, Sports, World*. Para la etapa de entrenamiento se ocupó un total de 480 artículos por sección, y para realizar las pruebas 120 noticias. El mejor resultado se obtiene utilizando *Multi-Variate Bernoulli Featureset* como algoritmo de extracción de características y *Naive Bayes Classification* como algoritmo clasificador ya que, obtiene un 77 % de exactitud.

2.3.3. Category classification and topic discovery of japanese and english news articles

Este trabajo desarrolla un algoritmo de aprendizaje supervisado (ver [Capítulo 3](#)) para la clasificación de noticias en categorías (como política, deportes, tecnología) y temas (sección de deportes: tenis, fútbol, golf) en diferentes

lenguajes, además se especializa en descubrir y clasificar temas emergentes en Internet (B. Bracewell et al., 2009). Se ocupa un método para extraer palabras claves en cualquier idioma propuesto por Bracewell (Bracewell et al., 2005), el cual obtiene palabras de muy alta calidad de un solo documento. Se definieron 8 secciones posibles a las que puede ser clasificado el artículo proporcionado, los cuales son:

- *Business*
- *Politics*
- *Crime and Misfortune*
- *Health*
- *Sports*
- *Entertainment*
- *Technology* y
- *Science and Nature*

Con ejemplos positivos el método entrena un clasificador para cada categoría. El proceso de clasificación consta de 4 pasos:

1. Las palabras claves son extraídas del documento dado
2. la probabilidad de pertenencia a cada categoría es calculado
3. Se crea un umbral de pertenencia dinámico
4. Finalmente se asigna el artículo a una categoría

Para desarrollo del método se implementó en lengua ingles y japones, se ocuparon 1,000 artículos descargados de sitios como Yahoo, de cada idioma. 800 se ocuparon en el entrenamiento y 200 para realizar pruebas.

Para contar con un punto de comparación se clasifico con algoritmos ya probados: Naive bayes, Árboles de decisión, Máxima entropía y el propuesto por el artículo. El mejor resultado fue dado por el método propuesto obteniendo 63.4 % en exhaustividad, 68.6 % en precisión y 65.9 % en la media-F.

2.3.4. Automatic news articles classification in indonesian language by using naive bayes classifier method

El artículo clasifica noticias ocupando el algoritmo clásico *Naive Bayes* ([Asyárie and Pribadi, 2009](#)). El método propuesto consiste en 3 tareas importantes: Pre-procesamiento el cual consiste en la siguiente serie de pasos:

1. *Case folding*: Proceso para convertir todas letras en minúsculas
2. *Parsing*: Es el proceso de convertir oraciones en palabras
3. *Stopwords elimination*: Es el proceso de eliminar palabras que se repiten con mucha frecuencia y no es información útil (Una definición mas amplia se da en el capítulo 3)
4. *Stemming*: Es un proceso de corte o eliminación de afijos en una palabra. Las variantes de los afijos son prefijos, sufijos, in-fijos y con-fijos (la combinación de prefijos y sufijos)

La segunda tarea es la etapa de entrenamiento del algoritmo y por último la clasificación de artículos. Cabe destacar que el método **Frecuencia de término** (Frecuencia de aparición de una palabra en un documento dado) es utilizado en la etapa de aprendizaje. Las secciones definidas en el trabajo son:

- *Economy*
- *Sport*
- *Tecnology*
- *Healt*
- *Metropolitan*

Para el proceso de aprendizaje se ocuparon 50 noticias por tópico, las cuales fueron recolectadas de los sitios web *Kompas*¹, *Republika*² y *Suara pamburuan*³. Las pruebas fueron realizadas con 12 noticias por sección. Además para

¹Sitio web Indu de noticias: <https://www.kompas.com>

²Sitio ya no disponible: <http://www.republika.com>

³Sitio web Indu: <https://sp.beritasatu.com>

tener una métrica en la eficiencia del método se calculó la precisión, exhaustividad y la media-F. Los resultados muestran que el método de Naive bayes es un clasificador con una media-F de 92.26 %.

2.3.5. News article text classification in indonesian language

Este documento busca el mejor algoritmo de clasificación en lenguaje Indu, comparando la eficiencia de algoritmos de selección de características (Palabras clave) y de clasificación de noticias (Wongso et al., 2017). Las secciones definidas por el artículo son las siguientes, *Economy, Health, Sports, Politic y Technology*; el trabajo realiza pre-procesamiento de datos con los métodos *lemmatization* y *Stopwords* para reducir el ruido en la información. Para la obtención de noticias se hace uso de la técnica *crawling*(ver Capítulo 3) en el sitio *ccnnindonesia*⁴. Se obtuvieron 1,000 artículos para cada sección. 800 se usaron para la etapa de entrenamiento y 200 para realizar pruebas. Se muestra la lista de los algoritmos implementados:

- Selección de características:
 - *Singular Value Decomposition*
 - *Term frequency-inverse document frequency*
- Clasificación:
 - *Support vector machine*
 - *Naïve bayes classifier*
 - *Gaussean naïve bayes*
 - *Multinomial naïve bayes*
 - *Multivariate naïve bayes*
 - *Bernulli naïve bayes*

El mejor resultado es en combinación de *Term frequency-inverse document frequency* y *Multinomial naïve bayes* con la precisión y exhaustividad mas alta el cual está alrededor de 98.4 % con un tiempo de 0.702 segundos, seguido de *Term frequency-inverse document frequency* y *Bernulli naïve bayes*(BNB) con 98.2 % en precisión y exhaustividad con un tiempo de .701 segundos.

⁴Sitio web de noticias: www.ccnnindonesia.com

2.4. Herramientas disponibles

Entre las herramientas de trabajo que son de utilidad para el procesamiento de lenguaje natural y aprendizaje automático se encuentran:

2.4.1. Cloud natural language

Google Cloud Natural Language ([Google, 2019](#)) revela la estructura y el significado del texto con modelos potentes de aprendizaje automático previamente entrenados en una API de REST fácil de usar y con modelos personalizados se puede utilizar para extraer información sobre personas, lugares, eventos y muchos otros datos, que se mencionan en documentos de texto, artículos periodísticos o entradas de blog. También se puede utilizar para comprender las opiniones sobre los productos expresadas en los medios sociales o analizar la intención en las conversaciones de los clientes que se den en un centro de atención telefónica o una aplicación de mensajería.

2.4.2. Googlebot

Es el crawler diseñado por Google para indexar el contenido nuevo o actualizado de Internet. Googlebot ([Google, 2018](#)) no sólo tiene la capacidad de rastrear e indexar los sitios web de Internet, sino que además puede extraer información de ficheros como pueden ser PDF, XLS, DOC, etc. Una vez el contenido está indexado, el servidor lo clasifica y establece un orden de relevancia para las distintas búsquedas que pueda efectuar un usuario, es decir, lo posiciona.

2.4.3. Watson natural language classifier

Watson NLC ([IBM, 2017](#)) aplica técnicas de computación cognitiva para analizar un texto y proporcionar la clase que mejor encaja entre un conjunto de clases predefinidas a partir de un texto corto. Al ser un clasificador, esta compuesto de ciertos pasos, en primera instancia se necesitan de clases las cuales son etiquetas que identificarán el texto analizado y será la salida proporcionada por el clasificador; posteriormente se debe tomar en cuenta que se

necesita de una colección de textos, los cuales proporcionarán apoyo para que el clasificador logre identificar las clases ingresadas posteriormente teniendo todos estos datos se logra entrenar al clasificador, el cual proporcionará una salida dependiendo a los datos que fueron utilizados.



Capítulo 3

Marco teórico



En este capítulo se expondrán de manera detallada y ordenada el conjunto de conocimientos que permitirán comprender y analizar el tema propuesto.

La Figura 3.1 muestra los campos abarcados por la investigación. A continuación cada área sera desarrollada con los conceptos de interés para la solución propuesta.

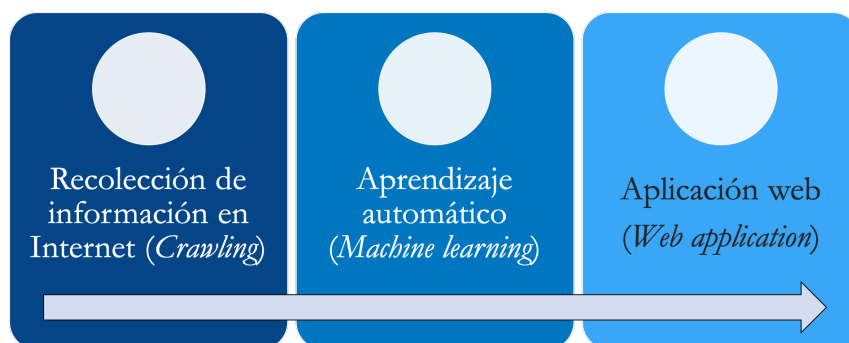


Figura 3.1: Campo de estudio

Aprendizaje automático

3.1. Inteligencia Artificial

Son muchas las definiciones que se encuentran de la inteligencia artificial o IA, en sus inicios se propone como las actividades asociadas al pensamiento humano, tareas como, toma de decisiones, resolución de problemas y aprendizaje (Bellman, 1978). Con el paso de los años se ha acuñado una definición mas completa: “la Inteligencia Artificial es una ciencia orientada al diseño y construcción de máquinas que implementen tareas propias de humanos dotados de inteligencia” (Pajares and Santos, 2006).

Esta ciencia contribuye en el desarrollo de diversos campos de investigación como, Redes neuronales, Computación evolutiva, Algoritmos genéticos, Programación Genética, Teoría del caos. Además tiene un campo amplio de aplicaciones en la sociedad (Russell and Norvig, 2009), a continuación se muestran algunos ejemplos:

- **Vehículos robóticos:** Un auto robótico sin conductor llamado STANLEY aceleró a través del terreno de Mojave a 22 mph (*miles per hour*, por sus siglas en ingles), terminando el curso de 132 millas primero para ganar el Gran Desafío DARPA 2005
- **Reconocimiento de voz:** Un viajero que llama a *United Airlines* para reservar un vuelo puede tener la conversación completa guiada por un sistema automático de reconocimiento de voz y gestión de diálogos
- **Planificación y programación autónoma:** A cien millones de millas de la Tierra, el programa *Remote Agent* de la NASA se convirtió en el primer programa autónomo de planificación a bordo para controlar la programación de operaciones de una nave espacial
- **Robótica:** *iRobot Corporation* ha vendido más de dos millones de aspiradoras robóticas *Roomba* para uso doméstico

- **Máquina traductora:** Un programa de computadora traduce automáticamente del árabe al inglés

3.2. Procesamiento de lenguaje natural

El procesamiento de lenguaje natural es una disciplina de la Inteligencia Artificial que se ocupa de la formulación e investigación de mecanismos computacionales para la comunicación entre personas y maquinas mediante el uso de Lenguajes Naturales.

Este campo incluye diferentes técnicas para interpretar el lenguaje humano, que van desde los métodos estadísticos y del aprendizaje basado en máquina hasta los enfoques basados en reglas y algorítmicos. Se necesita una amplia variedad de métodos porque los datos basados en texto y en voz varían ampliamente, al igual que las aplicaciones prácticas.

Dentro de la amplia gama de técnicas para el procesamiento de lenguaje natural, en este trabajo se harán uso de dos de estas técnicas llamadas **Tokenización** y **Lematización**. A continuación se describen cada una de ellas junto con un proceso previo llamado pre-procesamiento.

3.2.1. Pre-procesamiento

Cuando se recupera información de la web, se debe limpiar el texto ya que contiene etiquetas definidas por el hipertexto (ver [HTML](#)). Se deben buscar los bloques que brinden información útil para el ámbito de estudio, algunas secciones contienen publicidad o información no relacionada a los datos de interés. Para esto se tiene que realizar un análisis para discriminar la información útil ([Aggarwal, 2018](#)). A continuación se muestra un ejemplo de un texto que requiere ser pre-procesado

Cuadro 3.2.1: Texto de ejemplo

< h1 >¡Estudiantes politécnicos apoyan académicamente a alumnos de primaria!< /h1 >

∞ < p >El Secretario de Educación Pública 🇲🇽, Esteban Moctezuma Barragán, ha subrayado la importancia ### de que en la Nueva Escuela Mexicana el Instituto Politécnico Nacional (IPN) 😊, contribuya a fortalecer la educación básica.< /p >

<! – – – – >

Como se puede apreciar en el Cuadro 3.2.1 algunas etiquetas requieren ser descartadas para poder obtener únicamente el contenido de la noticias, por ejemplo se debe eliminar el *hipertexto* (< h1 >,< /p >) los emojis (😊,🇲🇽) y los símbolos (∞, #). El cuadro 3.2.2 muestra el resultado de limpiar el contenido.

Cuadro 3.2.2: Texto limpio

¡Estudiantes politécnicos apoyan académicamente a alumnos de primaria!

El Secretario de Educación Pública, Esteban Moctezuma Barragán, ha subrayado la importancia de que en la Nueva Escuela Mexicana el Instituto Politécnico Nacional (IPN), contribuya a fortalecer la educación básica.

3.2.1.1. Tokenización

Es el proceso que descompone los textos de una colección en sus unidades mínimas, las palabras o términos propiamente dichos. A tales elementos se les denomina tokens que conforman una lista de items que se utilizan para su análisis estadístico, lingüístico, de almacenamiento y posteriormente de recuperación de información. Los tokens a su vez pueden ser identificados mediante una codificación ASCII o en su defecto UNICODE. De hecho, este proceso permite la identificación de cadenas de caracteres de forma unívoca, de cara a posteriores tratamientos de depuración, eliminación de signos de

puntuación o la reducción morfológica (Blázquez, 2013).

Continuando con el ejemplo del Cuadro 3.2.2, se procede a realizar la tokenización del contenido. El Cuadro 3.2.3 muestra el texto dividido en tokens.

Cuadro 3.2.3: Texto tokenizado

¡ Estudiantes politécnicos apoyan académicamente a alumnos de primaria !

El Secretario de Educación Pública , Esteban Moctezuma Barragán , ha subrayado la importancia de que en la Nueva Escuela Mexicana el Instituto Politécnico Nacional (IPN) , contribuya a fortalecer la educación básica .

Como se puede observar se ha separado por un espacio las palabras, donde cada una representa un token. Además los signos de admiración (! ¡), los signos de puntuación (. ,) y los paréntesis (()), se han dividido como tokens independientes.

3.2.1.2. Lematización

Es el proceso lingüístico que, dada una palabra flexionada se encuentra su lema. Una palabra flexionada es cuando **esta** en plural, en femenino conjugada, diminutivo o en superlativo. El lema es la palabra que **esta** en singular para sustantivo, singular masculino para adjetivo e infinitivo para un verbo (Yunta, 2006). Ejemplo:

- amigos, amiga, amiguitos->Amigo
- soy, son, es->Ser

Cabe mencionar que existen diversos grados de lematización:

- Morfológica: Es la anteriormente explicada
- Sintáctica: Toma en cuenta el contexto donde se encuentra la palabra

En este trabajo se utilizó el grado morfológico. Continuando con el ejemplo tokenizado (Cuadro 3.2.3), El Cuadro 3.2.4 muestra el contenido del texto lematizado.

Cuadro 3.2.4: Texto Lematizado

¡ estudiante politécnicos apoyar academia a alumno de primaria !

el secretario de educar pública , esteban moctezuma barragán , haber
subrayar el importar de que en el nuevo escuela mexicano el instituto
politécnico nacional (ipn) , contribuir a fortaleza el educar básico .

3.3. Aprendizaje Automático

El Aprendizaje Automático es una rama de la Inteligencia Artificial; permite desarrollar algoritmos que tienen la capacidad de extrapolar (*i.e* predecir) los cambios que se acontecen en una tarea específica (Mueller and Massaron, 2016).

El campo utiliza una variedad de algoritmos que aprenden iterativamente de un conjunto de datos para describir y predecir resultados. A medida en la cual los algoritmos de entrenamiento obtienen datos es posible obtener modelos más precisos. Existen cuatro clasificaciones en los métodos (Marsland, 2014):

- **Aprendizaje supervisado:** Se proporciona un conjunto de datos de entrenamiento con las respuestas correctas y, con base a este conjunto de entrenamiento, el algoritmo genera un modelo para responder correctamente a todas las entradas posibles
- **Aprendizaje no supervisado:** No se proporcionan datos de entrenamiento, el algoritmo intenta identificar similitudes entre las entradas para clasificar en conjuntos. El enfoque estadístico del aprendizaje no supervisado se conoce como estimación de densidad

- **Aprendizaje reforzado:** Está en algún lugar entre el aprendizaje supervisado y no supervisado. Se indica al algoritmo cuando la respuesta es incorrecta, sin embargo no se informa cómo corregirlo. Tiene que explorar y probar diferentes posibilidades hasta que resuelva cómo obtener la respuesta correcta
- **Aprendizaje evolutivo:** La evolución biológica puede verse como un proceso de aprendizaje: los organismos biológicos se adaptan para mejorar sus tasas de supervivencia y la posibilidad de tener descendientes en su entorno. Este comportamiento es modelado, usando un modelo físico, el cual corresponde a una puntuación en la solución actual

Cabe mencionar que el método implementado en este trabajo es el aprendizaje supervisado, **más adelante se da una explicación detalla.**

El aprendizaje automático se puede aplicar a una amplia gama de problemas comerciales, desde la detección de fraudes hasta la orientación al cliente y la recomendación de productos, al monitoreo industrial en tiempo real, el análisis de sentimientos y el diagnóstico médico. Puede asumir problemas que no pueden administrarse manualmente debido a la gran cantidad de datos que deben procesarse (Fetherolf, 2016). Cuando se aplica a grandes conjuntos de datos, a veces puede encontrar relaciones tan sutiles que ninguna cantidad de escrutinio manual las descubriría nunca. Y cuando muchas de estas relaciones “débiles” se combinan, se convierten en predictores fuertes.

3.4. Aprendizaje automático para texto

La extracción de información útil con varios tipos de algoritmos estadísticos es denominado **Extracción de datos** (*text mining*), **Analítica de texto** (*text analytics*) o **Aprendizaje automático para texto** (*Machine learning for text*) (Aggarwal, 2018). En los últimos años este campo ha incrementado por el desarrollo de la web, redes sociales, correos electrónicos, bibliotecas virtuales. Algunas de las aplicaciones son las siguientes:

- Etiquetar la web, permite al usuario encontrar paginas de interés
- Los proveedores de correos, utilizan la información almacenada para mostrar publicidad de interés al usuario

- Algunas páginas ordenan su contenido de acuerdo a su importancia
- El análisis de las opiniones es un campo de importancia así como el análisis de sentimientos

El orden de las palabras en un texto brindan un significado semántico el cual no puede ser inferido solo con la frecuencia de las palabras. Sin embargo, se pueden hacer varias predicciones sin contemplar la semántica. Existen dos tipos de representaciones que son populares:

- **Texto como una bolsa de palabras:** Es la representación mas común. No se contempla el orden de las palabras en el proceso. El conjunto de palabras en el documento se convierten en una representación multidimensional dispersa, el cual corresponde a la dimensión en esta representación. Se utiliza para la clasificación, sistemas de recomendación
- **Texto como un conjunto de secuencias:** En esta representación se extraen sentencias, el orden de las palabras si importa. La unidad son sentencia o párrafos. Es utilizado en aplicaciones que necesitan un fuerte uso de la semántica, esta área se acerca mucho al modelado de lenguaje

3.4.1. Representación del texto

Los métodos de Aprendizaje Automático requieren que la información de la cual aprenderán esté representada en un formato que facilite su procesamiento. Generalmente esta representación es mediante vectores de valores numéricos. Cuando se requiere utilizar estos métodos con información en forma de texto, dicha información debe ser transformada para generar una representación más adecuada, los métodos mas comunes son: frecuencia, binaria y TF-IDF.

En este trabajo se utilizarán las dos primeras representaciones basandose en los resultados que han obtenido en el estado del arte. Por lo que sólo se describirán estas dos técnicas junto un ejemplo de su uso.

La extracción de características cuenta con dos tareas importantes: formar el vocabulario y crear un vector de características. Para ejemplificar esta tarea

observe el Cuadro 3.4.1 el cual es un corpus de 4 oraciones. Una vez realizado el proceso de extracción de características se obtiene el vocabulario el cual el mostrado en el Cuadro 3.4.2.

Cuadro 3.4.1: Corpus

$$\begin{bmatrix} \text{Este} & \text{es} & \text{el} & \text{primer} & \text{texto} & \text{!!} \\ \text{Este} & \text{texto} & \text{es} & \text{el} & \text{segundo} & \text{texto} & \text{???} \\ \text{Y} & \text{este} & \text{es} & \text{el} & \text{tercero} & & \\ \text{Es} & \text{este} & \text{el} & \text{primer} & \text{texto} & \text{????} \end{bmatrix}$$

Cuadro 3.4.2: Vocabulario

$$[\text{es} \text{ este} \text{ el} \text{ texto} \text{ primer} \text{ segundo} \text{ tercero} \text{ y} \text{ ?} \text{ !}]$$

Las características son extraídas de 2 formas, binario(donde 1 representa la presencia de la característica y 0 la ausencia) y por frecuencia (donde se cuenta el número de veces que cada característica aparece). Continuando con el ejemplo del Cuadro 3.4.1 se extraen las características por frecuencia y el resultado se muestra en el Cuadro 3.4.3, mientras que el Cuadro 3.4.4 muestra las características extraídas de forma binaria.

Cuadro 3.4.3: Representación por frecuencia

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 & 0 & 1 & 0 & 3 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 4 & 0 \end{bmatrix}$$

Cuadro 3.4.4: Representación binaria

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

3.5. Aprendizaje supervisado

Los algoritmos de aprendizaje supervisado dependen de datos previamente etiquetado, es decir se necesita un corpus de datos, para llevar a cabo el entrenamiento, así el algoritmo pueda comprender los datos y con ello determinar que etiqueta debe asignarse a los nuevos datos en función del patrón y asociando los patrones a los nuevos datos sin etiquetar. Después de ello, la máquina recibe un nuevo conjunto de datos para que el algoritmo de aprendizaje supervisado analice los datos y produzca un resultado correcto de los datos etiquetados ([CleverData, 2019](#)).

3.6. Clasificación multiclase

Existen dos tipos de clasificaciones: la clasificación binaria, donde se decide si un objetivo pertenece a una clase o no; y la clasificación multiclase en la cual, se tiene un conjunto de datos etiquetados y estos pertenecen a una de N clases diferentes. El objetivo en esta última es construir un algoritmo donde dado otro dato, este pueda predecir de forma correcta la clase a la cual pertenece el nuevo punto. A continuación se describen algunos de los métodos más utilizados de Aprendizaje Automático aplicados a tareas de texto.

3.6.1. Regresión logística

La regresión logística es una técnica estadística multivariante que nos permite estimar la relación existente entre una variable dependiente no métrica (donde la variable es binaria o también conocida como dicotómica, es decir, solo va a dar como resultado dos alternativas posibles) y un conjunto

de variables independientes métricas o no métricas (Velasco, 2002). Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores. El análisis de regresión logística se enmarca en el conjunto de Modelos Lineales Generalizados que usa como función de enlace la función logit. Las probabilidades que describen el posible resultado de un único ensayo se modelan, como una función de variables explicativas, utilizando una función logística.

Este algoritmo está basado en una regresión lineal, en el cual trata de optimizar la función l_1

$$\min_{w,c} |w|_1 + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1) \quad (3.1)$$

Otra forma de este clasificador es usando la función l_2 quien minimiza el costo de la función:

$$\min_{w,c} |w|_1 + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1) \quad (3.2)$$

La regresión logística es usada extensamente en las ciencias médicas y sociales. Otros nombres para regresión logística usados en varias áreas de aplicación incluyen modelo logístico, modelo logit, y clasificador de máxima entropía.

3.6.2. Naive bayes

Naive Bayes es un conjunto de algoritmos basados en el **teorema de Bayes** y el uso de la condición **Naive**. Generalmente utilizan aprendizaje supervisado sobre el conjunto de entrenamiento T para poder estimar los parámetros del modelo generativo, en tanto el conjunto de datos de entrada nuevos se realiza el teorema de Bayes, seleccionando la probable categoría que se ha generado (McCallum et al., 1998).

Usando la condición **Naive** todas las características extraídas que utilizan este clasificador se asumen independientes entre sí. La ventaja de usar este clasificador es que funciona bien tanto con datos numéricos como con datos textuales y, además, es más fácil de implementar. La desventaja de este clasificador es que su rendimiento empeora cuando las características extraídas

se correlacionan entre sí.

Una derivación de este algoritmo es llamada *Naive Bayes multinomial*, quien permite calcular la probabilidad de pertenencia de un texto d a una clase c , como se muestra en la siguiente ecuación:

$$P(c|d)\alpha P(c) \prod_{k=1}^n P(t_k|c) \quad (3.3)$$

donde:

- $P(c)$ Es la probabilidad de ocurrencia de una clase
- $P(t_k|c)$ Es la probabilidad condicional de aparición de una palabra en el conjunto de textos de c
- n Es el número de palabras en d

$$P(c) = \frac{N_c}{N} \quad (3.4)$$

donde:

- N_c Representa la cantidad de características (palabras) de c
- N Representa la cantidad total de características (es decir la unión de las palabras de cada clase)

$$P(t_k|c) = \frac{N_{ck} + \alpha}{N_c + \alpha n} \quad (3.5)$$

donde:

- $N_{ck} = \sum_{k \in T} t_k$ Es el número de veces que la característica k aparece en la clase c del corpus de entrenamiento T
- $N_c = \sum_{k=1}^n N_{ck}$ Es el número total de características que contiene la clase c
- n Es el número de características totales (es decir el vocabulario de la clase c_1, c_2, c_3)

Cabe destacar que la complejidad de este algoritmo es $\Theta(mc)$, donde m es el número de características por cada clase c .

3.6.3. Máquina de soporte vectorial

Las máquinas de soporte vectorial son sistemas de aprendizaje los cuales se basan en el uso de un espacio de funciones lineales en un espacio de mayor dimensión inducido por un kernel, en el que las hipótesis son entrenadas por un algoritmo (Suárez, 2014). Han sido implementadas en clasificación de imágenes, reconocimiento de caracteres, detección de proteínas, clasificación de patrones, identificación de funciones, etc. Pertenecen a la categoría de los clasificadores lineales, debido a que inducen separadores lineales (también conocidos como hiperplanos), ya sea en el espacio original de los ejemplos de entrada, si éstos son separables o cuasi-separables (ruido), o en un espacio transformado (espacio de características), si los ejemplos no son separables linealmente en el espacio original. La búsqueda del hiperplano de separación en estos espacios transformados, normalmente de muy alta dimensión, se hará de forma implícita utilizando las denominadas funciones kernel. Mientras la mayoría de los métodos de aprendizaje se centran en minimizar los errores cometidos por el modelo generado a partir de los ejemplos de entrenamiento (error empírico), el sesgo inductivo asociado a la SVM radica en la minimización del denominado riesgo estructural. La idea es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para, de esta forma, conseguir lo que se denomina un margen máximo a cada lado del hiperplano. Además, a la hora de definir el hiperplano, sólo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes.

3.6.4. Random forest

Random forest es una combinación de árboles de decisión, de modo que cada árbol depende de los valores de un vector aleatorio muestreado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de bagging que construye una larga colección de árboles no correlacionados y posteriormente los promedia (Breiman, 2001).

Bootstrap aggregating (bagging) consiste en obtener muestras aleatorias con reemplazamiento de igual tamaño que el conjunto original (Alfaro et al., 2003). Partiendo del conjunto de entrenamiento $X = (X_1, X_2, \dots, X_n)$, mediante la extracción aleatoria con reemplazamiento con el mismo número de elementos que el conjunto original de n elementos, se obtienen B mues-

tras bootstrap $X_b = (X_{1b}, X_{2b}, \dots, X_{nb})$ 11 donde $b=1, 2, \dots, B$. En algunas de estas muestras se habrá eliminado o al menos reducido la presencia de observaciones ruidosas, por lo que el clasificador construido en ese conjunto presentará un mejor comportamiento que el clasificador construido en el conjunto original. Así pues Bagging puede ser útil para construir un mejor clasificador cuando el conjunto de entrenamiento presente observaciones ruidosas.

La clasificación es realizada mediante votos, donde un **boto** se define como, la clasificación regresada por un árbol, la sección con el mayor número de votos es la clasificación asignada a los datos de entrada.

3.7. Métricas de evaluación de un modelo de aprendizaje automático

Una vez generando un modelo de clasificación, es importante medir el desempeño del mismo, con la intención de mejorar su eficiencia. Una de estas técnicas es la llamada matriz de confusión.

Matriz de confusión

Una matriz de confusión es una representación de la información de los resultados obtenidos por un clasificador, dicha matriz suele ser de tamaño $n \times n$, donde n es el número de clases diferentes con las que se están trabajando (Visa et al., 2011).

		Valor de predicción	
		Positivos	Negativos
Valor real	Positivos	Verdadero Positivo (VP)	Falso Negativo (FN)
	Negativos	Falso Positivos (FP)	Verdadero Negativo (VN)

Figura 3.2: Matriz de confusión

3.7. MÉTRICAS DE EVALUACIÓN CAPÍTULO 3. MARCO TEÓRICO

La Figura 3.2 muestra un ejemplo de matriz de confusión con dos clases, la cual ejemplifica de manera adecuada las diferentes entradas de la misma, entre las que se encuentran:

- **VP**: Es la cantidad de datos positivos que fueron clasificados correctamente como positivos
- **FN**: Es la cantidad de datos positivos que fueron clasificados incorrectamente como negativos
- **VN**: Es la cantidad de datos negativos que fueron clasificados correctamente como negativos
- **FP**: Es la cantidad de datos negativos que fueron clasificados incorrectamente como positivos

La diagonal principal en cualquier matriz de confusión $n \times n$ representa el número de predicciones correctas para cada una de las n secciones.

Gracias a la matriz de confusión, es posible obtener ciertas métricas que nos ayudan a evaluar el modelo de aprendizaje. Entre las que se encuentran:

Exactitud: es la proporción del número total de predicciones que son correctas respecto al total. Se determina utilizando la ecuación:

$$Exactitud = \frac{VP + VN}{VP + VN + FN + FP} \quad (3.6)$$

Recall: Es la proporción de predicciones positivas que fueron correctamente clasificadas. Se determina utilizando la ecuación:

$$Recall = \frac{VP}{VP + FP} \quad (3.7)$$

Precisión: Es la proporción de predicciones positivas que se clasificaron correctamente. Se determina con la siguiente ecuación:

$$Precision = \frac{VP}{VP + FN} \quad (3.8)$$

F-Measure (F1): Se interpreta como la media armónica entre Precisión y Recall. Se determina con la siguiente ecuación:

$$F - Measure = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3.9)$$

3.8. Validación cruzada

El proceso de validación cruzada es uno de los métodos mas usados para generalizar la capacidad de predecir de un modelo clasificador y para prevenir el sobreentrenamiento, ademas es usado en la etapa de entrenamiento de un algoritmo de aprendizaje supervisado ([Berrar, 2018](#)). Este método consiste en dividir el corpus en n pliegues como se muestra en la Figura 3.3, cada pliegue está conformado por **Dobleces** los cuales definen un conjunto de entrenamiento (doblez de color azul cielo) y otro de prueba (doblez de color azul rey). En cada pliegue se entrena el modelo y se prueba, para calcular la exactitud de este conjunto, al terminar el proceso se obtiene el promedio. El objetivo de la validación cruzada es estimar la exactitud del modelo en nuevos datos.

Cabe destacar que esta prueba permite encontrar un resultado más robusto y confiable en cuanto a la eficiencia del algoritmo, ya que asegura que los datos no sean manipulados para entrenar y probar con un conjunto de datos a conveniencia (es decir la combinación de información con la exactitud más alta).

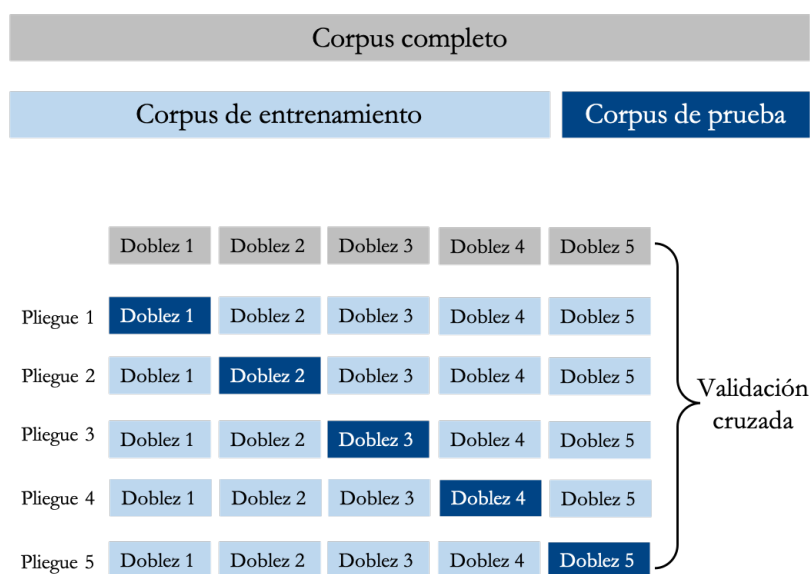


Figura 3.3: Validación cruzada

Recolección de información de Internet

3.9. Web Scraping

La recopilación de datos de Internet es una técnica que se realiza de manera manual, sin embargo el *Web Scraping* es el conjunto de técnicas utilizadas para obtener de manera automática información de un sitio web ([Vargiu and Urru, 2013](#)).

El *Web scraping* accede a las páginas web, encuentra los elementos de datos especificados en la página, los extrae y transforma en diferentes formatos si es necesario, finalmente, guarda la información como un conjunto de da-

tos estructurado¹. Los investigadores limpian y organizan el contenido para analizar la información.

3.9.1. Técnicas de web scraping

Algunas de las técnicas que nos proporciona el *Web scraping* son(Munzert et al., 2014a):

- **Copiar y pegar:** Realiza el método recolección copiar y pegar la información, sin embargo es una técnica propensa a errores
- **Uso de expresiones regulares:** Es una técnica que se puede utilizar para obtener la información de las páginas web son las expresiones regulares, aunque comúnmente no se recomienda utilizarlas para parsear el formato HTML
- **Reconocimiento de anotaciones semánticas:** Las páginas que contienen metadatos, marcas semánticas o explicaciones adicionales que se pueden usar para encontrar fragmentos de datos específicos
- **Parsers de HTML:** Algunos lenguajes, como XQuery y HTQL pueden ser utilizados para parsear documentos, recuperar y transformar el contenido de documentos HTML

En el presente trabajo se hará uso de la técnica de **expresiones regulares**. La Figura 3.4 muestra los procesos.

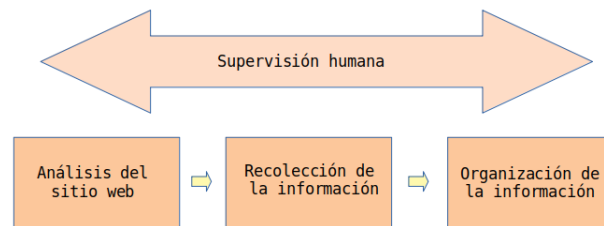


Figura 3.4: Etapas del proceso de *Web scraping*.

¹Un conjunto de datos estructurado permite recolectar varios valores simultáneamente.

3.10. Crawler

Un *Crawler* es una herramienta la cual analiza sitios web, permitiendo recolectar las páginas web para así posteriormente extraer la información que contengan (Alexey Grigorev and Reese, 2017). Un crawler también conocido como **como** robot o spider, es un sistema para la descarga masiva de páginas web. Son uno de los componentes principales de los motores de búsqueda web, los sistemas que reúnen un conjunto de páginas web, las indexan y permiten a los usuarios realizar consultas contra el índice y encontrar las sitios que coincidan con las consultas.

3.11. Python

*Python*² un lenguaje de programación creado en 1991, se ha convertido en uno de los más importantes lenguajes de programación para la ciencia de datos, el aprendizaje automático y el desarrollo general de software en el mundo académico y la industria. En los últimos años, el soporte mejorado de Python para bibliotecas (como pandas y scikit-learn) lo ha convertido en una opción popular para las tareas de análisis de datos. Combinado con la solidez general de Python para la ingeniería de software de propósito general, es una excelente opción como **idioma** principal para crear aplicaciones de datos (McKinney, 2012).

3.11.1. Scrapy

Scrapy es un *framework* para rastrear sitios web y extraer datos estructurados que pueden utilizarse para una amplia gama de aplicaciones útiles, como la extracción de datos, el procesamiento de información o el archivo histórico. A pesar de que Scrapy fue diseñado originalmente para el *Web scraping*, también se puede usar para extraer datos mediante API (como los Servicios web de Amazon Associates) (Kouzis-Loukas, 2016).

La arquitectura del proyecto Scrapy se basa en arañas, que son rastreadores independientes que reciben un conjunto de instrucciones, hace que sea más fácil construir y escalar grandes proyectos de *Crawler* al permitir que

²<http://www.python.org/>

los desarrolladores reutilicen su código. Scrapy también proporciona un terminal de rastreo web, que los desarrolladores pueden usar para probar sus suposiciones sobre el comportamiento de un sitio.



Aplicación Web

3.12. Aplicación Web

Las aplicaciones Web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil de usuario o el desarrollo de comercio electrónico.

3.13. Patrón de diseño

Para el análisis y desarrollo de aplicaciones es necesario seguir una técnica para tener el control de la aplicación, con el fin de darle mantenimiento. Se han desarrollado múltiples técnicas que cumplen con esta tarea.

3.14. Modelo Vista Controlador

El patrón deseable para el desarrollo de aplicaciones Web es el Modelo Vista Controlador (MVC), este modelo considera separar en tres capas un proyecto, permitiendo gestionar sistemas de software grandes y complejos, el cual puede ser implementado en sistemas Web.

- Lógica de control: saber qué elementos tiene el proyecto y qué hacer, pero no cómo se implementó
- Lógica de negocio: saber cómo se desarrolla la aplicación
- Lógica de presentación: saber cómo interactúa el usuario con la aplicación

El patrón Modelo Vista Controlador, es el más extendido para el desarrollo de aplicaciones donde se deben manejar interfaces de usuarios, éste se centra

en la separación de los datos o modelo, y la vista, mientras que el controlador es el encargado de relacionar a estos dos.

La Figura 3.5 muestra la separación de las tres capas.

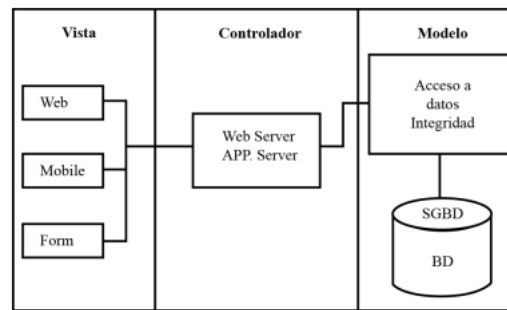


Figura 3.5: Patrón MVC Modelo Vista Controlador.

3.15. Framework

Un marco de trabajo (*Framework*) es una colección integrada de componentes, la cual simplifica el desarrollo de aplicaciones, al proporcionar bibliotecas y herramientas de software.

3.16. JavaServer Faces

JavaServer Faces (JSF) es un Framework para aplicaciones Web que simplifica el diseño de la interfaz de usuario y separa aún más la presentación de una aplicación Web de su lógica comercial.

JavaServer Faces proporciona dos bibliotecas de etiquetas personalizadas para agregar los componentes a una página. El programador diseña la apariencia visual de una página con JSF, agregando etiquetas

3.17. Internet

Son muchas las definiciones que se encuentran de la Internet, sin embargo su definición es simple, "la Internet es una colección de redes de comunicación interconectadas mediante el protocolo lo cual garantiza que las redes físicas que la componen, formen una red lógica única de alcance mundial"

(Musciano and Kennedy, 2002).

Uno de los principales objetivos cuando se desarrolló fue permitir la comunicación entre distintos usuarios, hoy en día es utilizado para muchos ámbitos, desde obtener información relevante, hasta comprar artículos provenientes de otros países.

El Instituto Nacional de Estadística y Geografía (INEGI) realizó una encuesta³ en donde se muestra que al año 2018 hay 18.3 millones de hogares que disponen de Internet, es decir más del 50 % de la población en México. El acceso a Internet ha crecido de manera exponencial, permitiendo a los usuarios tener infesar a distintos recursos.

3.17.1. World Wide Web

La WWW *World Wide Web* es un sistema de distribución de documentos HTML *HyperText Markup Language* que permite a los usuarios de computadora ejecutar aplicaciones basadas en Web, además de localizar y ver documentos basados en multimedia sobre casi cualquier tema a través de Internet. La *World Wide Web* es una colección gigante de documentos o páginas, almacenados en computadoras de todo el mundo. Comúnmente llamada la Web, esta colección de páginas representa una gran cantidad de texto, imágenes, audio y video disponibles para cualquier persona con una computadora y una conexión a Internet.

3.18. Hypertext markup language

HyperText Markup Language (*HTML*, por sus siglas en ingles), es un lenguaje que permite presentar contenido en la Web y fue propuesto por primera vez por Tim Berners-Lee (1989). El estándar ha evolucionado continuamente desde su introducción inicial, la versión más reciente es HTML5 que está siendo desarrollada por el *World Wide Web Consortium* (W3C).

Un archivo HTML es un texto sin formato, el cual se puede abrir y editar con cualquier editor de texto. Lo que hace al HTML tan poderoso es su estructura marcada, el cual permite definir las partes de un documento que

³<https://www.inegi.org.mx/programas/dutih/2018/>

deben mostrarse como titulares, las partes que contienen enlaces, las partes que deben organizarse como tablas y muchas otras formas. Las definiciones de marcado se basan en secuencias de caracteres predefinidas, las etiquetas, que encierran partes del texto ([Munzert et al., 2014b](#)).

3.19. Sitios Web

Un sitio Web es un conjunto de páginas Web relacionadas entre sí. Se entiende por página Web tanto el fichero que contiene el código HTML como todos los recursos que se emplean en la página, como pueden ser imágenes sonidos, videos ([McDaniel, 2011](#)). Un sitio Web son de acceso público que comparten un solo nombre de dominio, pueden ser creados y mantenidos por un individuo, grupo, empresa u organización para cumplir una variedad de propósitos. Todos estos sitios constituyen la World Wide Web.

3.19.1. Página Web

Una página Web es un documento electrónico el cual forma parte de la WWW (*World Wide Web*) generalmente construido en el lenguaje HTML (*Hyper Text Markup Language*). Este documento puede contener enlaces que nos direcciona a otra página Web. Para visualizar una página Web es necesario de un browser o un navegador. Dentro de las páginas Web se encuentra un sinfin de sitios los cuales pueden ser de interés. Las páginas Web pueden ser estáticas o dinámicas. Las páginas estáticas muestran el mismo contenido cada vez que se visualizan. Las páginas dinámicas tienen contenido que puede cambiar cada vez que se accede a ellas ([Marchal, 2001](#)).

3.19.2. Blog

Un blog es una página Web en la cual el usuario no necesita conocimientos específicos del medio electrónico ni del formato digital para poder aportar contenidos de forma inmediata, ágil y constante desde cualquier punto de conexión a Internet ([Bruguera, 2019](#)).

Un blog es un sitio Web que generalmente contiene información realizada por un autor. Esta información pueden ser de varios tipos, como comentarios, descripciones de eventos, fotos, videos, comentarios personales, tutoriales, estudios de casos, artículos de opinión extensos, ideas políticas o cualquier otra

cosa que pueda imaginar. Por lo general, se muestran en orden cronológico inverso, con las adiciones más recientes en la parte superior. Esas entradas de información se pueden organizar de varias maneras, por fecha, tema. Una de las características principales de un blog es que se debe actualizar periódicamente. A diferencia de un sitio Web donde el contenido es estático, un blog se comporta más como un diario en línea, donde el blogger publica actualizaciones periódicas. Por lo tanto, los blogs son dinámicos con contenido siempre cambiante. Un blog se puede actualizar con contenido nuevo y el contenido anterior se puede cambiar o eliminar en cualquier momento (aunque eliminar el contenido no es una práctica común) ([Krol, 2019](#)).

3.19.3. Foro

Un foro es una herramienta de comunicación asíncrona. Los foros permiten la comunicación de los participantes desde cualquier lugar en el que esté disponible una conexión a Internet sin que éstos tengan que estar dentro del sistema al mismo tiempo, de ahí su naturaleza asíncrona. Brindando una mayor interacción entre distintos participantes y permitiendo conocer la opinión sobre un tema de distintas personas. Los foros son probablemente el único recurso de resolución de problemas y recursos basados en información en la Internet, le brindan un entorno interactivo en el que puede aprender y aplicar sus conocimientos ([Mercer, 2006](#)).

Capítulo 4

Análisis y diseño



En este capítulo se describe el análisis y el diseño del sistema web para el trabajo terminal propuesto, mostrando los módulos con los cual cuenta. Hasta este punto se presentan los requisitos que deberá cumplir el sistema así como los casos de uso y diagramas de secuencia.

4.1. Actores y roles

Usuario: Cualquier persona que ingrese al sistema y esté interesada en consultar noticias.

4.2. Requisitos funcionales

RF1 Recolectar noticias



- **Descripción:** El sistema debe recolectar noticias de forma automática de los sitios web definidos.

RF2 Clasificar noticias



- **Descripción:** El sistema debe clasificar las noticias recolectadas de acuerdo a su contenido, en las secciones previamente definidas.

RF3 Filtrar noticias



- **Descripción:** El sistema debe filtrar las noticias recolectadas de acuerdo a la fecha de publicación; el periodo permitido para el filtrado de noticias es: de la fecha actual de ingreso al sistema hasta tres días antes. Cabe destacar que de encontrar noticias anteriores a este periodo, estas podrán ser visualizadas.

RF4 Mostrar resultados



- **Descripción:** El sistema debe mostrar las noticias que cumplan con los filtros de búsqueda establecidos por el usuario (Sección y fecha de publicación).

4.3. Requisitos no funcionales

RNF1 Tiempo de recolección y clasificación



- **Descripción:** El tiempo de recolección y clasificación de las noticias no debe tomar mas de 1 minuto.

RNF2 Número de palabras



- **Descripción:** Las noticias recolectadas deben tener un mínimo de 180 palabras en ellas.

RNF3 Número de noticias mostradas



- **Descripción:** El sistema debe mostrar al menos 5 noticias clasificadas, por página.

4.4. Reglas de negocio

En esta sección se describen las reglas de negocio implementadas en el trabajo propuesto.

RN1 Número de palabras



- **Descripción:** La noticia debe tener al menos 180 palabras
- **Referenciado por:** CU1 Recolectar noticias

RN2 Lenguaje de noticias



- **Descripción:** Las noticias deben estar redactadas en lenguaje español
- **Referenciado por:** CU2 Clasificar noticias

RN3 Listado de fuentes noticiosas



- **Descripción:** Solo se puede recolectar información de los siguientes sitios
 - **El Universal:** <https://www.eluniversal.com.mx/>
 - **Azteca Noticias:** <https://www.aztecanoticias.com.mx/>
 - **Aristegui Noticias:** <https://aristeguinoticias.com/>
 - **La Jornada:** <https://www.jornada.com.mx/ultimas>
 - **Sopitas:** <https://www.sopitas.com/>
 - **El Economista:** <https://www.eleconomista.com.mx/>
 - **Proceso:** <https://www.proceso.com.mx/>
- **Referenciado por:** [CU1 Recolectar noticias](#)

RN4 Número de noticias recolectadas



- **Descripción:** El número máximo de noticias recolectadas por sitio web debe ser 30
- **Referenciado por:** [CU1 Recolectar noticias](#)

RN5 Orden de publicación



- **Descripción:** Las noticias se muestran con base a la fecha de publicación
- **Referenciado por:** [CU1 Recolectar noticias](#), [CU4 Mostrar resultados](#)

RN6 Periodo de recolección



- **Descripción:** De cada sitio establecido se recolectan las noticias que se encuentren en un periodo de al menos 3 días anterior a la fecha actual
- **Referenciado por:** CU1 [Recolectar noticias](#)

RN7 Campos recolectados de noticia



- **Descripción:** De cada noticia se extrae **Título**, **URL al artículo**, **Fecha de publicación** y de contar con ello el **Resumen**
- **Referenciado por:** CU1 [Recolectar noticias](#)

RN8 Periodo de actualización



- **Descripción:** El proceso de recolección de noticias se hará en periodos de 4 horas
- **Referenciado por:** CU1 [Recolectar noticias](#)

4.5. Casos de uso

4.5.1. Diagrama de casos de uso

La Figura 4.1 muestra el diagrama de casos de uso de la aplicación.

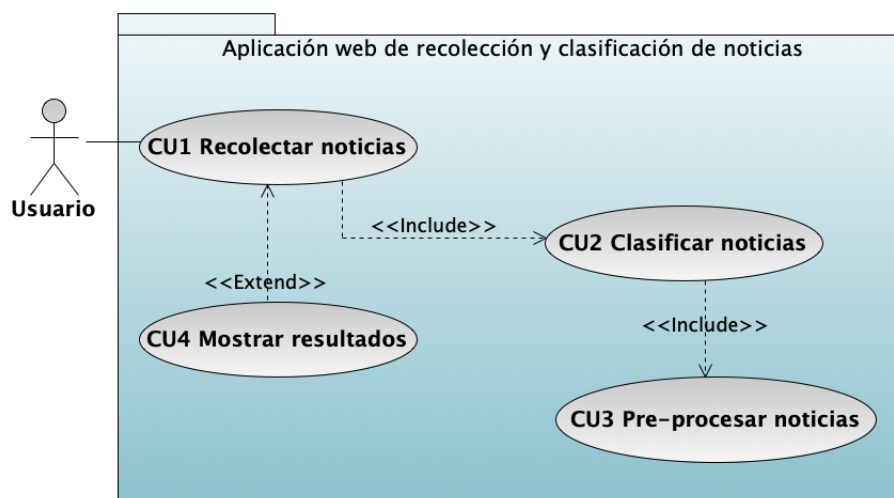


Figura 4.1: Diagrama de casos de uso

4.5.2. CU1 Recolectar noticias








Resumen

Brinda al usuario un punto de acceso para elegir una sección; las clasificaciones son, **Ciencia y tecnología**, **Política**, **Deportes**, **Economía** y **Cultura**, posteriormente se recolectan noticias de la web, tomando como punto de partida los sitios establecidos previamente. De cada sitio se recolectan las noticias publicadas; de cada artículo se obtiene **Fecha de publicación**, **Título**, **Contenido**, **URL de la noticia**, y de contar con ello el **Resumen**.

Descripción



Caso de uso:	CU1 Recolectar noticias
Actor:	Usuario
Propósito:	Brindar una herramienta de recolección de noticias de Internet(Crawler)
Entradas:	URL de las paginas por consultar
Salidas:	MSG1 Tiempo de recolección excedido
Precondición:	Tener un punto de conexión a Internet
Postcondiciones:	<ul style="list-style-type: none"> • El usuario tendrá la facultad de visualizar las noticias clasificadas • El usuario podrá cambiar el periodo de búsqueda
Reglas de negocio:	<ul style="list-style-type: none"> • RN1 Número de palabras • RN3 Listado de fuentes noticiosas • RN5 Orden de publicación • RN6 Periodo de recolección • RN7 Campos recolectados de noticia • RN8 Periodo de actualización
Errores:	<ul style="list-style-type: none"> • Uno: Cuando el tiempo de recolección se ha excedido se muestra el mensaje MSG1 Tiempo de recolección excedido

Trayectoria principal

1.  Selecciona una opción de la pantalla **UI1 Inicio; Política, Economía, Deportes, Ciencia y tecnología o Cultura**.
2.  Verifica que no existan noticias recolectadas previamente. [Trayectoria A]
3.  Muestra la pantalla **Pantalla UI2 Espera de proceso**.
4.  Por cada sitio se extraen las noticias con base en la regla de negocio RN3Listado de fuentes noticiosas, RN6 Periodo de recolección y RN7 Campos recolectados de noticia. [Trayectoria D]
5.  Incluye el caso de uso **CU2 Clasificar noticias**.
6.  Se obtienen las noticias clasificadas de la sección seleccionada por el usuario, de acuerdo a la regla de negocio RN5 Orden de publicación.
7.  Muestra la pantalla **Pantalla UI3 Proceso concluido**.
8. - - - - *Fin del caso de uso.*


Trayectoria alternativa A:

Condición: *Existen noticias recolectadas*

- A-1.  Verifica que la última recolección de noticias no exceda el periodo establecido, con base RN8 Periodo de actualización .
- A-2.  Continúa en el paso 6 de la trayectoria principal.
- A-3. - - - - *Fin de la trayectoria.*

Trayectoria alternativa B:

Condición: *La última recolección de noticias excede el periodo establecido*

- B-1.  Continúa en el paso 3 de la trayectoria principal.
- B-2. - - - - *Fin de la trayectoria.*

Puntos de extensión

Causa de la extensión: El usuario desea consultar las noticias clasificadas.

Región de la trayectoria: Proviene del paso 6 de la trayectoria principal.

Extiende a : [CU4 Mostrar resultados](#)

4.5.3. CU2 Clasificar noticias





Resumen

Brinda al sistema una herramienta que permite realizar la clasificación de las noticias recolectadas, en las secciones **Ciencia y tecnología**, **Política**, **Deportes**, **Economía** y **Cultura**, utilizando como modelo clasificador el algoritmo **Máquina de Soporte Vectorial**. Además el conjunto de noticias clasificadas es almacenado en un archivo por cada sección. Cabe señalar que las descargas se hacen en un periodo establecido.

Descripción

Caso de uso:	CU1 Recolectar noticias
Actor:	Usuario
Propósito:	Clasificar las noticias recolectadas
Entradas:	<ul style="list-style-type: none">• Noticias recolectadas• Modelo clasificador
Salidas:	Los archivos de cada sección los cuales contienen el conjunto de noticias correspondientes
Precondiciones:	Debe existir al menos una noticia recolectada
Postcondiciones:	Las noticias clasificadas podrán ser obtenidas por el sistema
Reglas de negocio:	Ninguna
Errores:	Ninguno

Trayectoria principal

1.  Obtiene las noticias recolectadas.
2.  Incluye el caso de uso **CU3 Pre-procesar noticias**.
3.  Obtiene el vocabulario definido, para el modelo clasificador.
4.  Generará un vector de características por cada noticia, con base al vocabulario del paso 3.

5. ● Obtiene el modelo clasificador.
6. ● Clasifica las noticias recolectadas.
7. ● Almacena las noticias clasificadas por sección.
8. - - - Fin del caso de uso.

4.5.4. CU3 Pre-procesar noticias

Resumen

Realiza dos tareas fundamentales, previas al proceso de clasificación las cuales son: **Tokenizar**. Este proceso consiste en dividir el texto en sus elementos mínimos llamados tokens, donde se separan palabras, signos de puntuación, llaves y números mediante un espacio; **Lematizar**. Esta tarea reduce cada palabra en su lema, con el objetivo de simplificar el vocabulario de un texto. Es importante mencionar que, de cada artículo se extrae la **URL**, **Título**, **Fecha** **Contenido de la noticia**, y de existir un **Resumen**. Sin embargo para el proceso de clasificación solo se utiliza la redacción de la noticia.

Descripción

Caso de uso:	CU1 Recolectar noticias
Actor:	Usuario
Propósito:	Preparar el contenido de las noticias para el proceso de extracción de características
Entradas:	Contenido de los artículos recolectados
Salidas:	Contenido procesado de los artículos
Precondición:	Vocabulario en español para el proceso de lematización
Postcondiciones:	Se podrá generar el vector de características de cada noticia
Reglas de negocio:	Ninguna
Errores:	Ninguno

Trayectoria principal

1. ● Obtiene el contenido de las noticias.
2. ● Realiza el proceso de tokenización.
3. ● Realiza el proceso de lematización.
4. - - - *Fin del caso de uso.*

4.5.5. CU4 Mostrar resultados





Resumen

Permite al actor visualizar las noticias correspondiente a la sección elegida, ya sea **Política**, **Deportes**, **Ciencia y tecnología**, **Economía** o **Cultura**. Cada artículo contiene el **Título**, **Autor**, la **Fecha de publicación**, **URL** el cual direcciona a la página fuente que ha proporcionado la noticias y de contar con ello un **Resumen de la noticia**. Además se permite filtrar las noticias en un periodo de al menos tres días. Cabe destacar que las noticias mostradas por defecto, corresponden a la fecha actual.

Descripción



Caso de uso:	CU4 Mostrar resultados
Actor:	Usuario
Propósito:	Brindar una herramienta que permita consultar las noticias clasificadas y filtradas por su fecha de publicación
Entradas:	Periodo de búsqueda
Salidas:	<ul style="list-style-type: none"> • MSG2 Petición vacía • Noticias clasificadas; de cada una se muestra: <ul style="list-style-type: none"> ◦ Título ◦ URL al artículo ◦ Fecha de publicación ◦ Autor ◦ Resumen
Precondición:	Las noticias clasificadas deben estar almacenadas por sección
Postcondiciones:	Ninguna
Reglas de negocio:	RN5 Orden de publicación
Errores:	Uno: Cuando no se ha encontrado noticias en el día seleccionado se muestra el mensaje MSG2 Petición vacía

Trayectoria principal

1.  Presiona el botón **Continuar** de la pantalla [Pantalla UI3 Proceso concluido](#). [[Trayectoria A](#)]
2.  Obtiene la fecha actual.
3.  Muestra al menos 5 noticias, de acuerdo a la regla de negocio [RN5 Orden de publicación](#) de la sección previamente elegida (filtro de sección) y del día actual (filtro de fecha), como se visualiza en la pantalla [UI4 Resultados de consulta](#).
4.  Consulta la noticia. [[Trayectoria B](#)]
5. - - - - *Fin del caso de uso.*




Trayectoria alternativa A:

Condición: *El usuario ha presionado el botón cancelar*

- A-1.  Presiona el botón **Cancelar** de la pantalla [Pantalla UI3 Proceso concluido](#).
- A-2.  Muestra la pantalla [UI1 Inicio](#).
- A-3. - - - - *Fin de la trayectoria.*

Trayectoria alternativa B:

Condición: *El usuario ha cambiado el periodo establecido*

- B-1.  Presiona un botón del menú **Cambio de periodo** de la pantalla [UI4 Resultados de consulta](#).
- B-2.  Verifica que exista al menos 1 noticia en el periodo establecido. [[Trayectoria C](#)]
- B-3.  Muestra noticias de acuerdo a la regla de negocio [RN5 Orden de publicación](#) de la sección previamente elegida (filtro de sección) y del día seleccionado en el paso 1 (filtro de fecha), como se visualiza en la pantalla [UI5 Cambio de periodo](#).

B-4. ● Continúa en el paso 4 de la trayectoria principal.

B-5. - - - - *Fin de la trayectoria.*

Trayectoria alternativa C:

Condición: *No se ha encontrado noticias del día seleccionado [Error Uno]*

C-1. ● Muestra el mensaje MSG2 Petición vacía, en la pantalla UI4 Resultados de consulta.

C-2. ● Continúa en el paso 4 de la trayectoria principal.

C-3. - - - - *Fin de la trayectoria.*

4.6. Mensajes

MSG1 Tiempo de recolección excedido



- **Tipo:** Error.
- **Objetivo:** Dar a conocer que se ha excedido el tiempo de recolección de noticias de los sitios web definidos
- **Redacción:** Se ha agotado el tiempo de espera, no se han encontrado noticias. Intentar más tarde
- **Referenciado por:** [CU1 Recolectar noticias](#)

MSG2 Petición vacía



- **Tipo:** Error
- **Objetivo:** Informar al usuario que no se ha encontrado resultados en el día seleccionado
- **Redacción:** No se ha encontrado noticias en el día seleccionado
- **Referenciado por:** [CU4 Mostrar resultados](#)

4.7. Pantallas

4.7.1. UI1 Página de inicio

Objetivo

Permite al usuario seleccionar la sección a consultar.

Descripción

La Pantalla [4.2](#) muestra un menú con las secciones definidas para el sistema las cuales son: **Política**, **Deportes**, **Ciencia y tecnología**, **Economía** y **Cultura**. En ella se puede navegar para acceder a la consulta de las noticias recolectadas y clasificadas de alguna sección.

Salidas

- [MSG2 Tiempo de recolección excedido](#)

Comandos

1. **Inicio:** Direcciona a la pantalla descrita en esta sección
2. **Política:** Permite realizar una consulta en la sección Política
3. **Deportes:** Permite realizar una consulta en la sección Deportes
4. **Ciencia y tecnología:** Permite realizar una consulta en la sección Ciencia
5. **Economía:** Permite realizar una consulta en la sección Economía
6. **Cultura:** Permite realizar una consulta en la sección Cultura

Referenciado por

- [CU1 Recolectar noticias](#)
- [CU4 Mostrar resultados](#)



Figura 4.2: Pantalla UI1 Inicio

4.7.2. UI2 Recolección y clasificación

Objetivo

Informar al usuario que la recolección y clasificación de noticias se está llevando a cabo.

Descripción

La Pantalla 4.3 muestra un mensaje flotante con la siguiente redacción: **En proceso de recolección y clasificación**, para informar al usuario que la consulta realizada está en proceso.

Salidas

- Ninguno

Referenciado por

- [CU1 Recolectar noticias](#)



Figura 4.3: Pantalla UI2 Espera de proceso

4.7.3. UI3 Proceso concluido

Objetivo

Informar al usuario que la recolección y clasificación ha concluido y brinda un punto de acceso para visualizar los resultados de la consulta.

Descripción

La Pantalla 4.4 muestra un mensaje flotante con la siguiente redacción: **No-ticias listas para ser mostradas**, el cual informa al usuario que el proceso de recolección y clasificación ha concluido, *i.e* ya se pueden mostrar las noticias clasificadas de la sección elegida. En la parte inferior se muestra el botón **Cancelar** y **Continuar**.

Salidas

- Ninguno

Comandos

1. **Cancelar**: Detiene el proceso de recolección y clasificación, re-direcciona

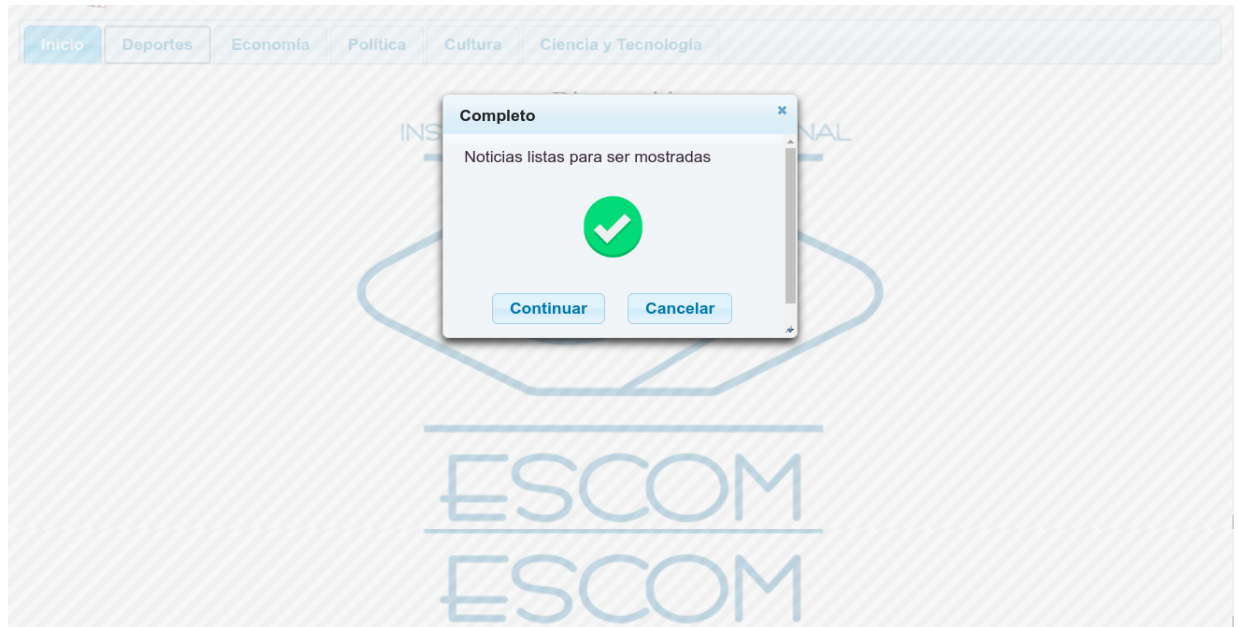


Figura 4.4: Pantalla UI3 Proceso concluido

a la pantalla [UI1 Inicio](#).

2. **Continuar:** Permite avanzar para visualizar las noticias clasificadas de la sección elegida en la pantalla [UI4 Resultados de consulta](#)

Referenciado por

- [CU1 Recolectar noticias](#)
- [CU4 Mostrar resultados](#)

4.7.4. UI4 Resultados de consulta

Objetivo

Permite consultar las noticias clasificadas en la sección previamente elegida. Además brinda una forma de cambiar el periodo de búsqueda y permite entrar al sitio de origen de los artículos mostrados.

Descripción

La Pantalla 4.5 muestra una sección con las noticias clasificadas, de cada noticia se muestra:

- **Título**
- **URL al artículo**
- **Fecha de publicación**
- **Resumen**

En la parte superior de la pantalla se muestra el menú **Cambio de periodo** el cual permite cambiar le periodo de consulta de las noticias. Cabe señalar que la primera vez que se ingresa a esta pantalla se muestran los artículos con fecha de publicación del día actual. La Pantalla 4.6 muestra un ejemplo de consulta en una fecha diferente.

Salidas

- [MSG2 Petición vacía](#)

Comandos

1. **Hoy:** Realiza la consulta en la fecha actual
2. **Ayer:** Realiza la consulta un día antes de la fecha actual
3. **Dos días:** Realiza la consulta dos días antes de la fecha actual
4. **Tres días:** Realiza la consulta tres días antes de la fecha actual
5. **URL:** La url que muestra la noticia direcciona al sitio web de recolección

Referenciado por

- [CU4 Mostrar resultados](#)



Figura 4.5: Pantalla UI4 Resultados de consulta



Figura 4.6: Pantalla UI5 Cambio de periodo

4.8. Diagrama de secuencia

La figura 4.7 muestra el diagrama de secuencia de la aplicación.

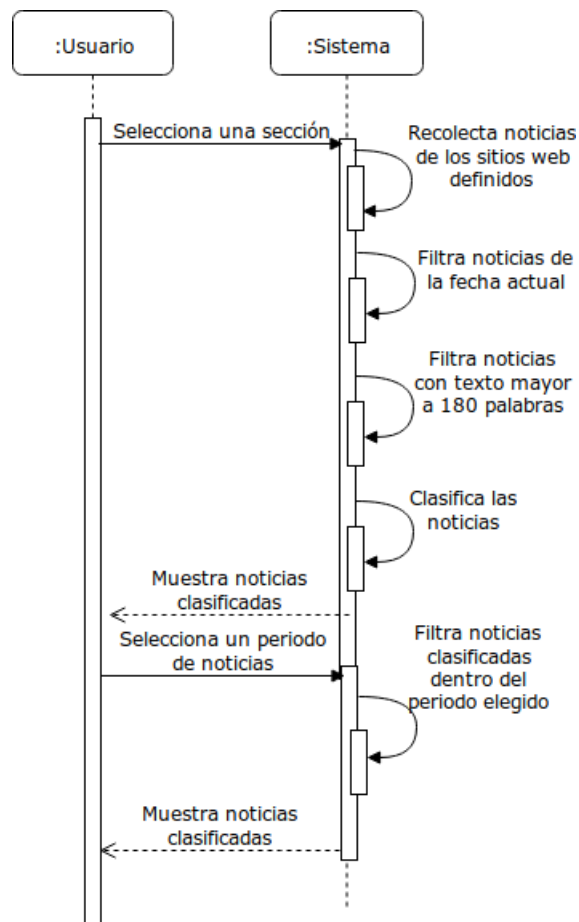
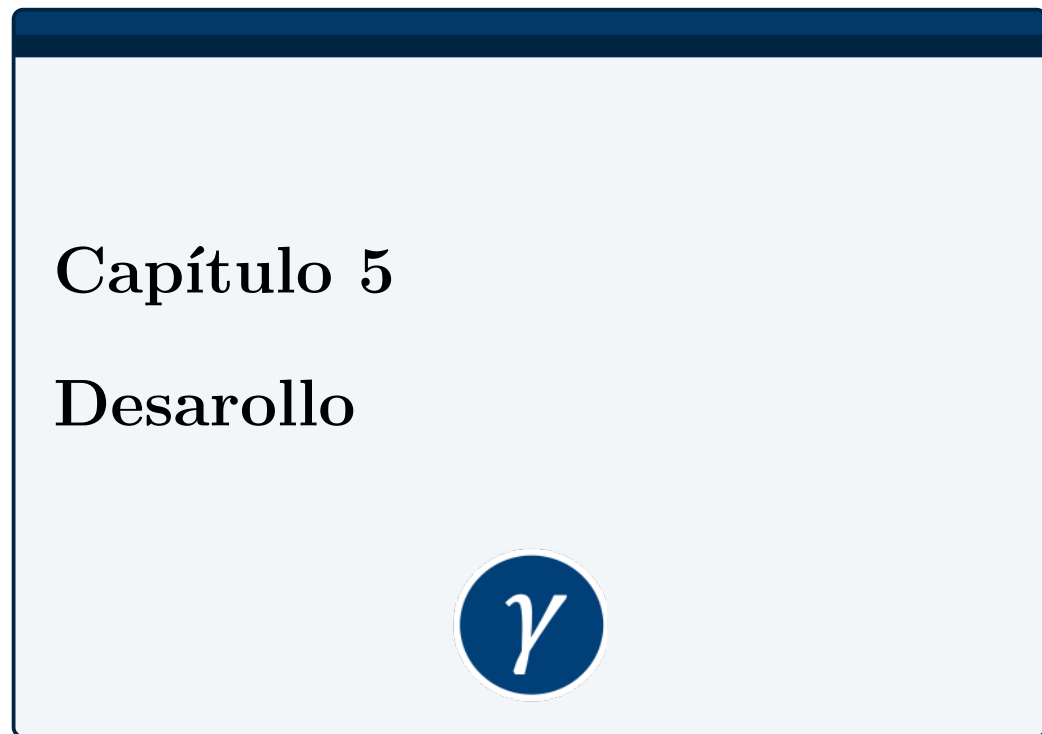


Figura 4.7: Diagrama de secuencia



El propósito de este capítulo es describir cada etapa de desarrollo del presente trabajo. Las etapas son mostradas en la Figura 5.1. Además se incluyen los resultados obtenidos en las pruebas realizadas.



Figura 5.1: Etapas de desarrollo

5.1. Recolección

El proceso de recolección es parte fundamental del presente trabajo terminal, ya que permitió conformar el corpus utilizado en la etapa de entrenamiento, la Figura 5.2 muestra las etapas que se desarrollaron durante el proceso de recolección.



Figura 5.2: Etapas de la recolección

5.1.1. Selección de sitios web

*Reuters Institute*¹ realizó un informe anual para comprender como se consumen las noticias en distintos países, mediante el sitio *YouGov*² se realizó la investigación utilizando cuestionarios en línea durante finales de enero y principios de febrero del año 2019. El informe reportó el índice de confianza que tienen los usuarios sobre los sitios noticiosos, la Figura 5.3 muestra los sitios que tienen un mayor índice de confianza en una escala de 0 a 10.

¹<https://reutersinstitute.politics.ox.ac.uk/>

²<https://mx.yougov.com/>

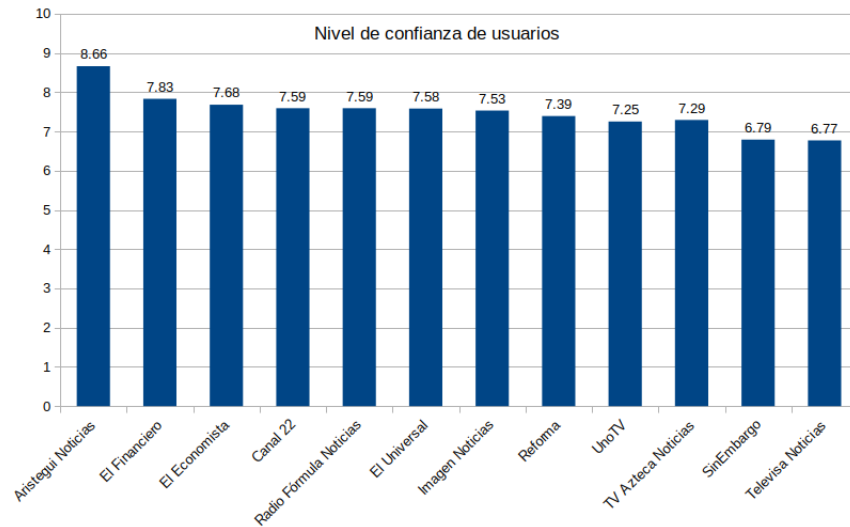


Figura 5.3: Nivel de confianza de usuarios al consultar un sitio web.

El sitio web El Economista³ contiene una sección llamada **Ranking de Medios Nativos Digitales**⁴, el cual muestra las estadísticas que realiza mes con mes acerca de los sitios de noticias web más consultados como se muestra en la Figura 5.4.

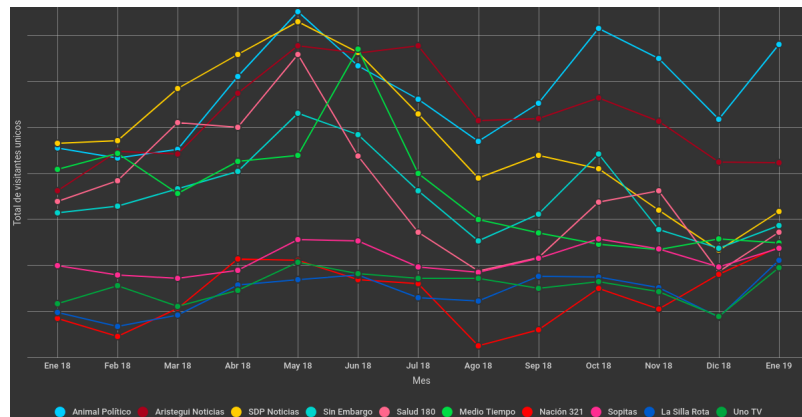


Figura 5.4: Ranking de sitios de noticias del período de enero del 2018 a enero del 2019.

³<https://www.eleconomista.com.mx/>

⁴<https://www.eleconomista.com.mx/Ranking-de-Medios-Nativos-Digitales>

La compañía *comScore*⁵ dedicada a proporcionar datos de marketing en Internet, en el año 2018 realizó un estudio en el cual indica los sitios web de noticias más visitados, los resultados se muestran en la Figura 5.5.

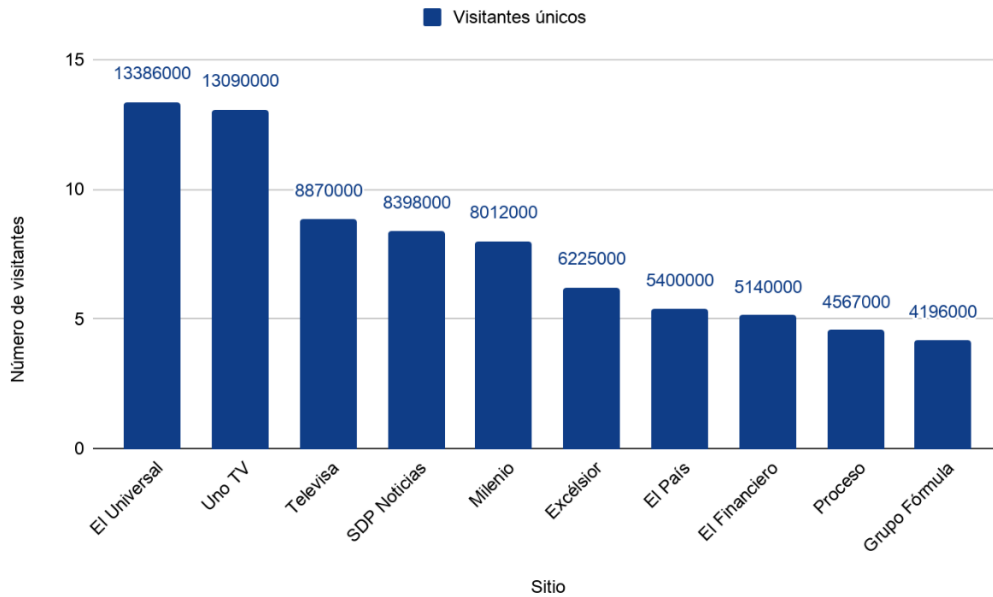


Figura 5.5: Visitantes únicos a sitios de noticias durante el año 2018.

Para llevar a cabo la selección de sitios web utilizados en el presente proyecto se realizó un análisis sobre aquellos que proveen de información noticiosa, entre los cuales se consideraron **foros de noticias**, **sitios de diarios** y **sitios de televisión**, en conjunto con las Figuras 5.3, 5.5, 5.4 se han seleccionado los sitios siguientes sitios:

- Aristegui Noticias
- El Economista
- La Jornada
- La Prensa
- Proceso

⁵<https://www.comscore.com/>

■ Sopitas y TV Azteca

De estas fuentes se recolectan noticias, ya que se desea obtener noticias de diversas fuentes de información, como lo son foros, blogs y sitios de televisoras.

Una vez definidos los sitios, observamos que clasifican las noticias por secciones, lo cual permite una búsqueda más rápida de la información, la Tabla 5.1. muestra el análisis realizado a las secciones contenidas en los sitios seleccionados.

Sitios	Secciones									
	Nacional	Internacional	Ciudad	Estados	Economía	Deportes	Espectáculos	Cultura	Política	Ciencia y tecnología
Aristegui Noticias	México	Mundo	-	México	Economía	Deportes	-	-	Poderes	-
Azteca Noticias	-	Internacional	-	Estados	Finanzas	Deportes	Entretenimiento	-	Política	Geek
El Economista	Úrbes y Estados	The Washington Post	Úrbes y Estados	Úrbes y Estados	Valores y Dinero	DxT	-	Artes, Ideas y Gente	Política y Sociedad	Tecnología
La Jornada	-	Mundo	CDMX	Estados	Economía	Deportes	Espectáculos	Cultura	Política	Tecnología
La Prensa	México	Mundo	Metrópoli	República	-	Deportes	Gossip	Cultura	México	Tecnología
Proceso	Nacional	Internacional	La Capital	Estados	-	Deportes	Miscelánea	Cultura	Política	Tecnología
Sopitas	Noticias	-	-	-	-	Deportes	En el show	-	-	Geek

Tabla 5.1: Secciones existentes en los sitios web

Una vez que se analizaron las secciones con las que contaba cada sitio, se procedió a homologar las secciones en las cuales la mayoría de los sitios coincidían, por lo cual se quedarón definidas 5 secciones para clasificación de las noticias:

- **Política**
- **Deportes**
- **Ciencia y tecnología**
- **Economía**
- **Cultura**

Cabe mencionar que hay muy pocos sitios que publican noticias de la sección **Ciencia y tecnología** y **Cultura**, por ello se consideraron éstas secciones.

5.1.2. Análisis de sitios web

Una vez definida la información requerida de cada noticia se realizó un análisis sobre la estructura **XML** (*Extensible Markup Language*, por sus siglas en inglés), con el fin de realizar expresiones *XPath* que permiten recorrer y procesar un documento XML, dado que cada sitio web cuenta con una estructura diferente, ha sido necesario realizar el análisis individual. Cabe mencionar que existen sitios los cuales realizan actualizaciones a su página, por esta razón cada dos meses se analizaban, con el fin de verificar que la estructura XML no cambiará.

Una expresión *XPath* de ruta permite buscar y seleccionar los distintos nodos de un documento XML. En el siguiente Cuadro 5.1.1 se muestra un ejemplo con los elementos de una nota, los cuales son: **para**, **de**, **título**, **texto**, en un documento XML estos son los nodos que conforman una nota.

Cuadro 5.1.1: Documento XML

```
<nota>
  <para>Daniel</para>
  <de>Andres</de>
  <título>Recordatorio</título>
  <texto>Recuerda despertar temprano.</texto>
</nota>
```

La expresión *XPath* que permite extraer el contenido de la etiqueta **<texto></texto>** se muestra en el Cuadro 5.1.2:

Cuadro 5.1.2: Expresión XPath

```
/nota/texto/text()
```

Para cada sitio web se crearon expresiones *XPath* para recolectar el contenido.

5.1.3. Creación de recolector

Como se explicó en el capítulo 3 (ver [Crawler](#)) un *Crawler* permite descargar información de una página web, como se muestra en la Figura 5.6. La im-

plementación en el trabajo terminal ha requerido diseñar 7 recolectores, uno por cada sitio web (ver [sitios web](#)).

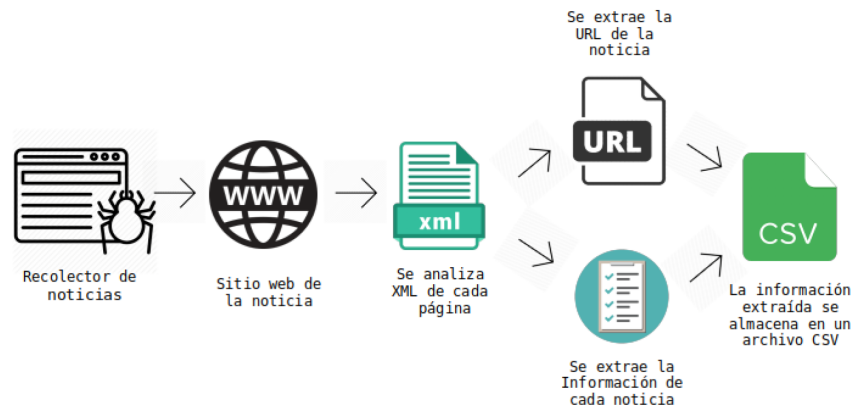


Figura 5.6: Proceso de recolección

El desarrollo del presente trabajo terminal se ha realizado en sistema operativo *Linux* en su distribución Ubuntu, para realizar la creación de los recolectores se ha utilizado el lenguaje de programación **Python 3**⁶, en conjunto con **Scrapy**⁷, Framework que permite la extracción de información de sitios web.

La información que ha sido recuperada de las noticias se muestra a continuación:

- **URL:** La dirección web donde se encuentra localizada la noticia
- **Título:** Encabezado de la noticia recolectada
- **Autor:** Es el nombre de la persona que redactó la noticia o el nombre de la editorial
- **Fecha:** Es la fecha en la cual la noticia ha sido publicada
- **Descripción:** Es una idea general del contenido de la noticia. Cabe mencionar que no todas las noticias cuentan con una descripción

⁶<https://www.python.org/>

⁷<https://scrapy.org/>

- **Noticia:** Es la redacción realizada por el autor acerca de la noticia. Es de relevancia mencionar que este elemento más importante de los artículos decargados

Cada uno de los recolectores contenía expresiones *XPath* que permitían recolectar la información de cada noticia, el Cuadro 5.1.3 muestra un ejemplo de las expresiones *XPath* utilizadas para recolectar noticias del sitio web Aristegui Noticias⁸

Cuadro 5.1.3: Ejemplo de expresiones *XPath* del sitio Aristegui Noticias

```
url= url
titulo = //div[@class="class_subtitular"]/h1/text()
autor = //div[@class="share_nom"]/text()
fecha = //div[@class="share_publicado"]/text()
descripcion = //div[@class="class_text2"]/text()
noticia = //div[@class="class_text"]/p/child::node()/text()
```

Cabe destacar que las noticias recolectadas se almacenaron en un archivo **CSV** (*comma separeted values*, por sus siglas en inglés), con la estructura que se muestra en la Tabla 5.2, donde la primera fila (Encabezado) define los elementos de este archivo, además las filas consecuentes representan el contenido recolectado de cada noticia.

url	título	autor	fecha	descripción	noticia
url ejemplo 1	título ejemplo 1	autor ejemplo 1	fecha ejemplo 1	descripción ejemplo 1	noticia ejemplo1
url ejemplo 2	título ejemplo 2	autor ejemplo 2	fecha ejemplo 2	descripción ejemplo 2	noticia ejemplo2

Tabla 5.2: Ejemplo de estructura de un archivo CSV

5.1.4. Recolección de noticias

Para el desarrollo de esta etapa, se recolectaron noticias de las secciones: **ciencia y tecnología, cultura, deportes, economía y política**, de los sitios web **Aristegui Noticias, El Economista, La Jornada, La Prensa, Proceso, Sopitas y TV Azteca**, durante el periodo de julio a septiembre

⁸<https://aristeguinoticias.com/>

del año 2019, cada cuatro días, con el fin de no tener noticias repetidas. El almacenamiento de las noticias se realizó en un directorio por sección, dentro de cada uno de estos se guardaban las noticias recolectadas por sitio web.

Una vez finalizada la primera etapa de recolección, los resultados obtenidos por sección se muestran en la Figura 5.7.

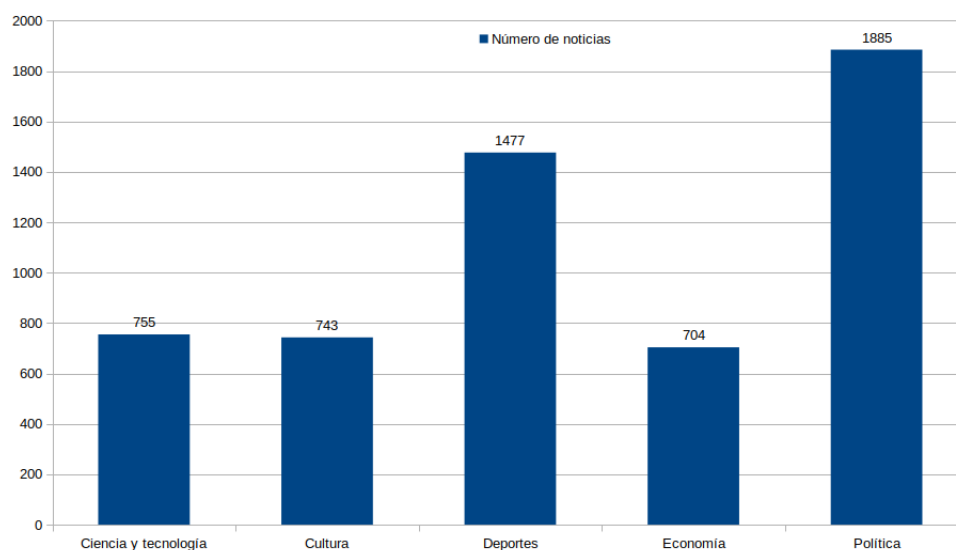


Figura 5.7: Noticias recolectadas durante el primer corte.

Cabe destacar que el número de noticias recolectado durante el primer corte, no se encontraba balanceado, es decir el número de noticias por sección era distinto, por ello se decidió continuar con el proceso de recolección de noticias, con el fin de balancear el corpus.

Una vez finalizada la segunda etapa de recolección el número de noticias se muestra en la Figura 5.8

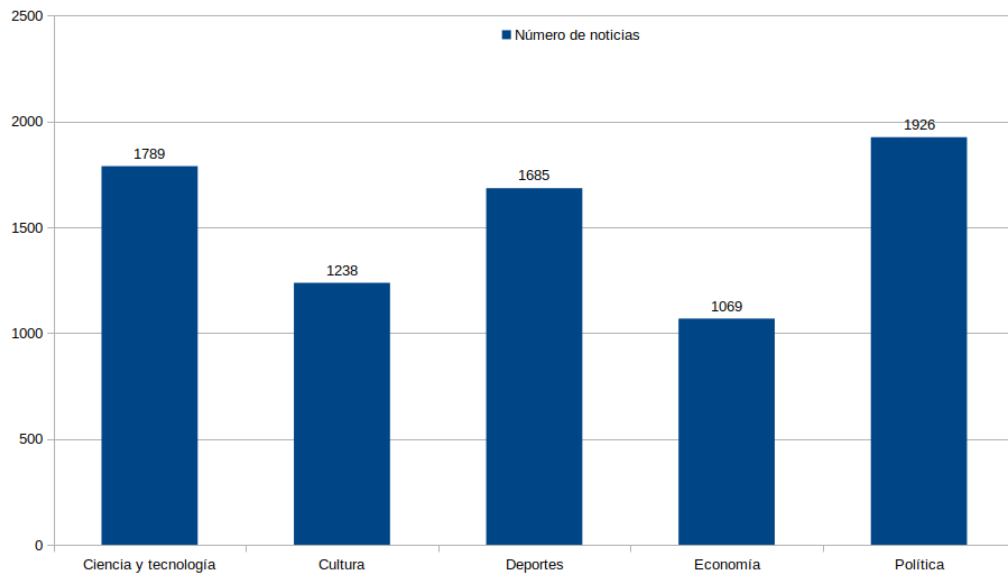


Figura 5.8: Noticias recolectadas al finalizar el segundo corte.

Los resultados que obtuvimos por sitio web se muestran en la Figura 5.9

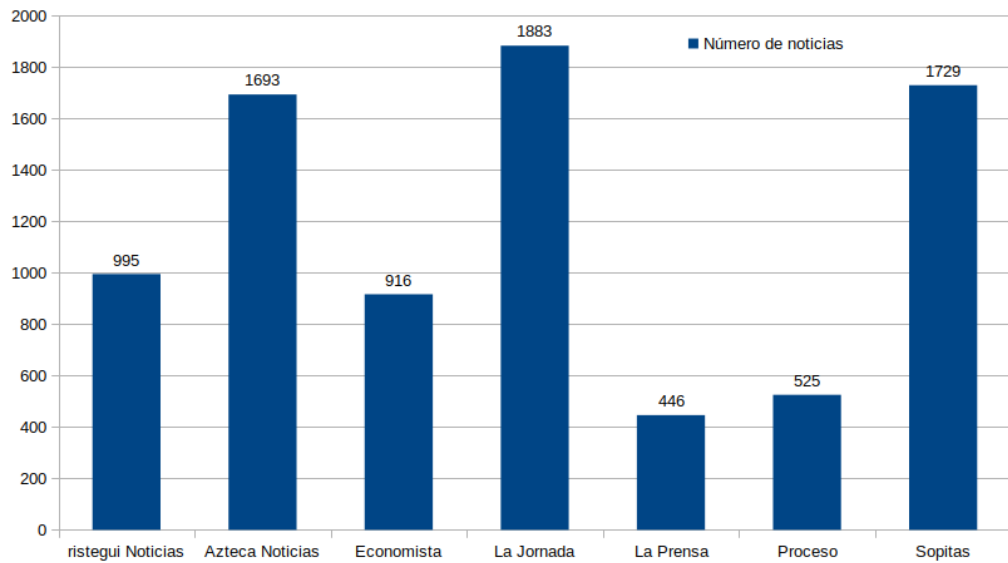


Figura 5.9: Noticias recolectadas por sitio web al finalizar el segundo corte

Una vez concluida la recolección de noticias, se eliminaron aquellas noticias que se encontraban más de una vez dentro de los archivos, utilizando listas en **Python** en el cual se validaba la URL de la noticia no se encontrará en la lista. Cabe mencionar que existían noticias que contenían *tuits* redactados en lenguaje inglés, por esta razón se procedió a eliminar el contenido de este idioma. Al concluir este proceso se obtuvo un total de 7,707 artículos. La Figura 5.10 muestra el total de noticias por sección y la Tabla 5.3 muestra las noticias recuperadas por cada sitio web.

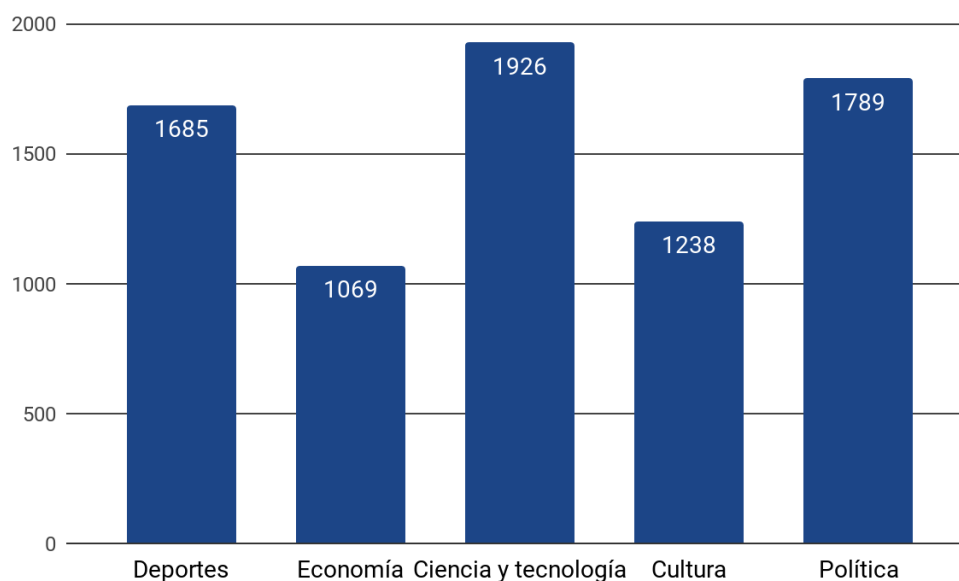


Figura 5.10: Noticias recolectadas secciones

Sitio	Sección	Número de Noticias
Aristegui Noticias	Ciencia y Tecnología	99
	Cultura	179
	Deportes	308
	Economía	161
	Política	248
Azteca Noticias	Ciencia y Tecnología	986
	Cultura	0
	Deportes	280
	Economía	77
	Política	350
El Economista	Ciencia y Tecnología	18
	Cultura	267
	Deportes	214
	Economía	201
	Política	236
La Jornada	Ciencia y Tecnología	4
	Cultura	424
	Deportes	284
	Economía	512
	Política	659
La Prensa	Ciencia y Tecnología	68
	Cultura	90
	Deportes	93
	Economía	118
	Política	77
Proceso	Ciencia y Tecnología	65
	Cultura	13
	Deportes	335
	Economía	0
	Política	112
Sopitas	Ciencia y Tecnología	549
	Cultura	265
	Deportes	171
	Economía	0
	Política	244

Tabla 5.3: Número de noticias recolectadas por sección de los sitios web

5.2. Entrenamiento de clasificador

El segundo pilar del trabajo terminal es entrenar un algoritmo (de aprendizaje automático), para clasificar las noticias en las secciones **Ciencia y tecnología**, **Política**, **Deportes**, **Economía** y **Cultura**. Cabe destacar que el clasificador resolverá un problema multiclase (ver [Clasificación multiclase](#)) debido a que la entrada es una artículo y como salida brinda la pertenencia a una sección de 5 posibilidades. El proceso de entrenamiento se muestra en la Figura 5.12.

En este punto es importante mencionar que existe un trabajo terminal previo (TT 2017-A042) el cual ha generado un modelo para clasificar noticias por secciones, sin embargo existen diferencias importantes entre el TT 2017-A042 y el propuesto en este documento, las cuales son mostradas en la Figura 5.11.

	<u>2017-A042</u>	<u>2018-B013</u>
01	Clasificación de noticias de diarios de circulación nacional mediante aprendizaje automático	Recolector y clasificador de noticias
02	Clasificación de 3 diarios nacionales	Clasificación de 7 sitios web
03	Se tiene un clasificador especializado a cada diario	El clasificador es general
04	Las noticias son ingresadas al sistema mediante teclado	Las noticias son ingresadas al sistema de forma automática

Figura 5.11: Diferencias del clasificador en el TT 2017-A042 y 2018-B013

Como se puede observar en la Figura 5.11 el modelo del TT 2017-A042 está orientado en la clasificación de noticias de 3 diarios de circulación nacional (**El Universal**, **La Jornada** y **Excélsior**), además de estar enfocado en las secciones particulares de cada periódico y de haber entrenado un algoritmo por cada diario. Por otra parte, este trabajo terminal (TT 2018-B013) tiene como objetivo clasificar noticias en 5 secciones (**deportes**, **economía**, **política**, **cultura**, **ciencia y tecnología**), de 7 diferentes sitios web. Es importante mencionar que se entrenará un solo modelo el cual generalice la tarea de clasificación de todas las fuentes noticiosas.

Dada las diferencias se decidió utilizar un nuevo corpus **como** entrenar un nuevo clasificador que se ajuste mejor al objetivo de este trabajo terminal.

Como se mencionó en la etapa de recolección, el corpus obtenido contiene 7,707 noticias en total. Sin embargo este contenido debe ser procesado y solo seleccionar los artículos que aporten información para el entrenamiento, al final de la etapa de preprocesamiento el corpus será dividido en dos conjuntos: entrenamiento con el 90 % de los artículos y prueba con el 10 %. A continuación se explica la etapa de **Pre-procesamiento**.

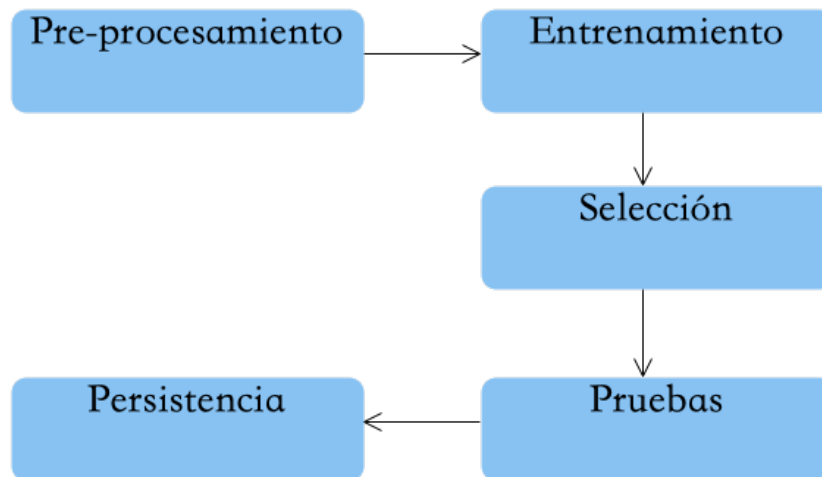


Figura 5.12: Proceso de entrenamiento

5.2.1. Preprocesamiento

Como primera instancia, el corpus creado debe ser procesado, con el fin de crear vectores que representan el contenido de cada artículo de forma ordenada (ver [Representación del texto](#)), de esta manera los algoritmos de clasificación son capaces de entender la información. En cuanto a los datos que son procesados de la noticia cabe mencionar que solo se usa el título y la redacción del artículo, los demás datos (como url, fecha, sección) no son necesarios para el entrenamiento. Este proceso consta de 6 etapas, las cuales son mostradas en la Figura 5.13.

Cabe destacar que estas etapas están desarrolladas en un script escrito en el lenguaje **python 3**, en cada sección se hará mención de las bibliotecas ocupadas.

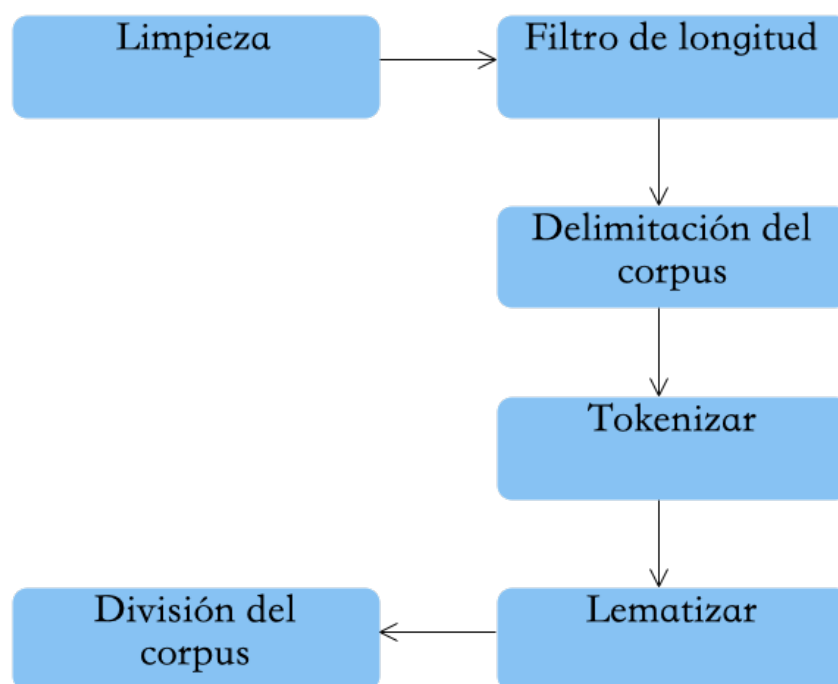


Figura 5.13: Etapas de preprocesamiento

5.2.1.1. Limpieza

Esta etapa consiste en eliminar texto que no brinda información útil para el entrenamiento como, hipertexto (ver [HTML](#)), símbolos especiales (como # † √), *emojis* (como 😊 😞 🍷). Por ejemplo el texto 5.2.1 muestra la redacción de una noticia con la información descargada de una pagina web. El resultado de limpiar la noticia se muestra en el texto 5.2.2. Se puede observar que se han eliminado los elementos `< p >` `< /p >` `< ! - - - - >` `# † 😊 😞 🍷`.

Para realizar esta tarea se han utilizado las bibliotecas: **pandas** como medio de lectura de archivos tipo **CSV** (*comma separeted values*, por sus siglas en ingles); **re** la cual permite evaluar expresiones regulares con el objetivo

de eliminar símbolos; **demoji** quien permite eliminar *emojis* dentro del texto.

Cuadro 5.2.1: Texto de entrada

< p > † El número 343 de El Trimestre Económico,⊗ revista emblemática del Fondo de Cultura <!-- -- -- > Económica (FCE), será * *
* presentado por David Ibarra Muñoz 😊, Carlos Tello Macías 😊, Alicia Puyana 😊 y Pablo Ruiz Nápoles el martes 27 de agosto a las 6 de la tarde, en la librería Rosario Castellanos, ubicada en avenida Tamaulipas # 202, en la colonia Condesa de la capital mexicana.< /p >

Cuadro 5.2.2: Texto limpio

El número 343 de El Trimestre Económico, revista emblemática del Fondo de Cultura Económica (FCE), será presentado por David Ibarra Muñoz, Carlos Tello Macías , Alicia Puyana y Pablo Ruiz Nápoles el martes 27 de agosto a las 6 de la tarde, en la librería Rosario Castellanos, ubicada en avenida Tamaulipas 202, en la colonia Condesa de la capital mexicana.

5.2.1.2. Filtro de longitud

Con base en la regla de negocio [RN1 Número de palabras](#) se ha definido 180 palabras como longitud mínima de las noticias, incluyendo en la definición de palabra números, signos de puntuación y exclamación. Para esto después de haber concluido el proceso de limpieza se pregunta por la longitud de la cadena (sin contar los espacios) y si esta es mayor o igual a 180, entonces es un artículo válido para utilizar en el entrenamiento de lo contrario no es tomado en cuenta.

5.2.1.3. Delimitación del corpus

En este punto del proceso, la cantidad de artículos ha disminuido (como se observa en la Figura [5.14](#)), sin embargo la distribución por sección no es homogénea, esta situación crea un sesgo en el corpus y después en la etapa

de **Entrenamiento** los algoritmos pueden desarrollar una preferencia por la categoría con mas datos. Por está razón se debe acotar la cantidad de noticias.

Como se muestra en la Figura 5.14 cada sección contiene al menos 700 noticias, por lo tanto este es el número definido para delimitar el número de artículos. De esta manera el corpus se ha definido como se muestra en la Figura 5.15, obteniendo un total de 3,500 noticias.

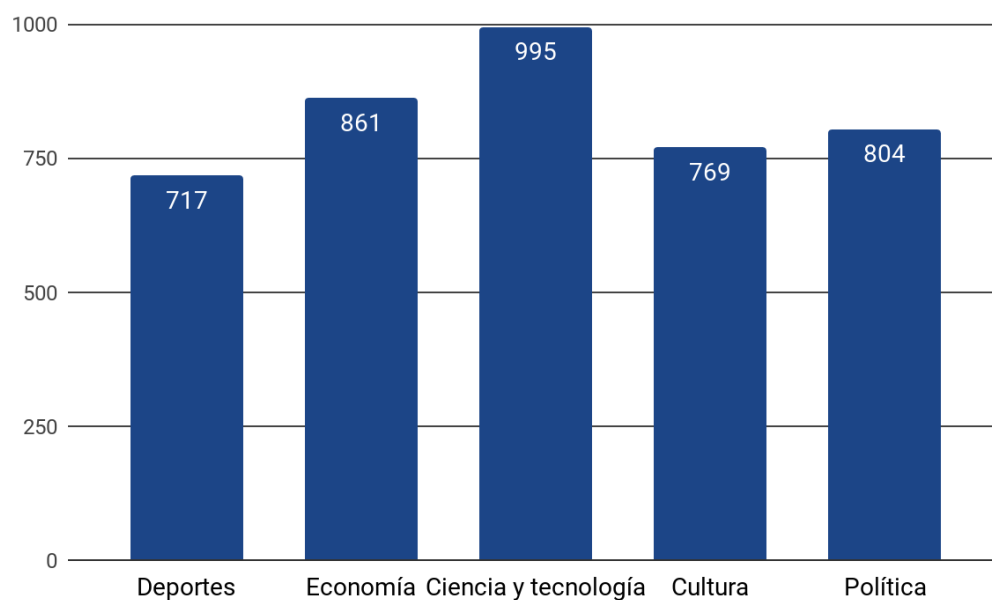


Figura 5.14: Noticias por sección

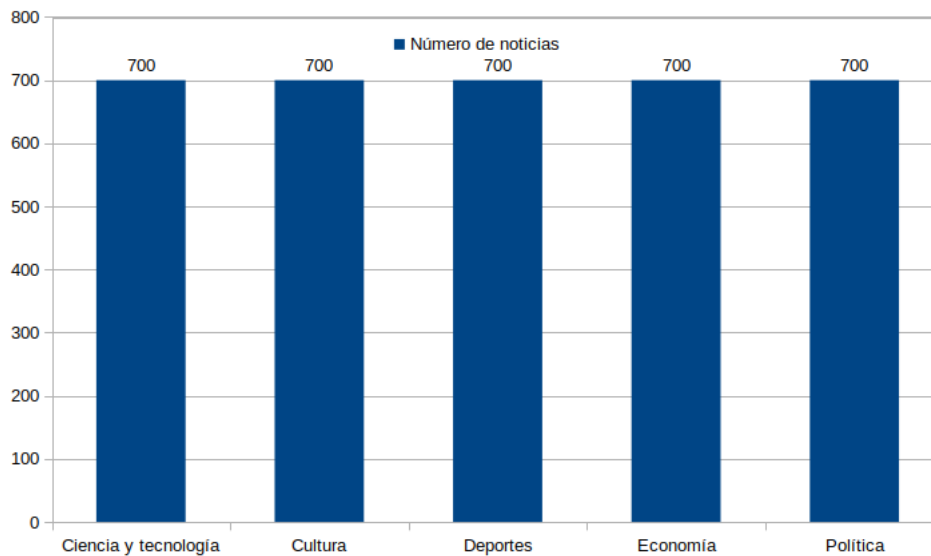


Figura 5.15: Noticias por sección

5.2.1.4. Tokenizar

La etapa de tokenización consiste en separar el texto en sus elementos mínimos llamados tokens, donde se separan palabras, signos de puntuación, llaves y números mediante un espacio. Continuando con el ejemplo 5.2.2 donde el texto se encuentra limpio, se procede a su tokenización. El resultado es mostrado en el Cuadro 5.2.3. Para remarcar el ejemplo observe la palabra entre paréntesis (**FCE**) la cual es separada en (**FCE**) mostrando que ahora cada elemento representa un token individual. Para el desarrollo de esta tarea se utilizó la biblioteca **RegexTokenizer**.

Cuadro 5.2.3: Texto tokenizado

El número 343 de El Trimestre Económico , revista emblemática del Fondo de Cultura Económica (FCE) , será presentado por David Ibarra Muñoz , Carlos Tello Macías , Alicia Puyana y Pablo Ruiz Nápoles el martes 27 de agosto a las 6 de la tarde , en la librería Rosario Castellanos , ubicada en avenida Tamaulipas 202 , en la colonia Condesa de la capital mexicana .

5.2.1.5. Lematizar

Lematizar es el proceso de reducir cada palabra a su lema, con el fin de disminuir la dispersión en el texto, por ejemplo las palabras correrás, corriendo, corrí, tienen como lema el verbo correr, el plural niños tiene como lema niño (ver [Lematización](#)). Para realizar esta tarea se ha usado **spacy** el cual es una librería de código abierto, con el diccionario *es_core_news_sm* quien permite analizar el léxico del lenguaje español.

Siguiendo con las etapas del proceso se toma el texto [5.2.3](#) como entrada al programa y este da como salida el texto que se muestra en el Cuadro [5.2.4](#).

Cuadro 5.2.4: Texto lematizado

el número 343 de el trimestre económico , revista emblemático del fondo de cultura económica (fce) , ser presentar por david ibarra muñoz , carlos tello macías , alicia puyana y pablo ruiz nápoles el martes 27 de agostar a los 6 de lo tardar , en lo librería rosario castellanos , ubicar en avenir tamaulipas 202 , en lo colonia condesa de lo capital mexicano .

Cuando el proceso de lematización concluye se genera un identificador único para cada noticia el cual se define de la siguiente forma

$$id = < \text{Identificador de sitio web} > < \text{Numero de noticias} >$$

donde **Identificador de sitio web** define un número único para hacer referencia a los sitios web (ver Tabla [5.4](#)) y **Número de noticias** es el número del artículo.

Como segundo paso las noticias son almacenadas en un archivo con extensión **TXT**, los elementos por almacenar son **id**, **título**, **noticia** y **sección**, los cuales son separados por los caracteres &&&&. El Cuadro [5.2.5](#) muestra un ejemplo de la estructura del archivo.

Número	Página web
100	Aristegui noticias
200	Tv azteca
300	El economista
400	La jornada
500	La prensa
600	Proceso
700	Sopitas

Tabla 5.4: Identificador de sito web

Cuadro 5.2.5: Estructura de archivo

id&&&&titulo&&&¬icia&&&&seccion
 1001&&&&Titulo 1&&&&Contenido noticia 1&&&&0
 ...
 5003500&&&&Titulo 3500&&&&Contenido noticia 3500&&&&4

5.2.1.6. División del corpus

Para el correcto diseño y evaluación del algoritmo clasificador se requiere dividir el corpus en dos conjuntos: **entrenamiento** y **prueba**, con un 90 % y 10 % del total del corpus respectivamente. En cada grupo deben estar repartidas noticias de las 5 secciones definidas, sin embargo los artículos almacenados están ordenados de forma descendente como: **Deportes**, **Economía**, **Política**, **Cultura**, **Ciencia y tecnología**, para seleccionar de forma distribuida los datos se ha utilizado una técnica llamada *Shuffle*.

Shuffle consiste en brindar un arreglo con los identificadores de las noticias y un número (nombrado usualmente como semilla), quien generar un nuevo orden en los identificadores de acuerdo a los números pseudo aleatorios que retorna esta función, tomando este como el nueva orden de los textos en el archivo de almacenamiento. Para el desarrollo de esta etapa se ha utilizado la biblioteca **Shuffle** con una semilla de 5.

Para ilustrar un ejemplo, en el cuadro 5.2.6 se muestra un conjunto de identificadores ordenados por el último número del *id*, al utilizar la función *Shuffle*

con una semilla de 2, se genera un nuevo orden el cual es mostrado en el Cuadro 5.2.7.

Cuadro 5.2.6: Identificadores de noticias

[1001 2002 3003 4004 5005 6006 7007 1008]

Cuadro 5.2.7: Nuevo orden de ID's

[5005 2002 7007 3003 4004 1008 6006 1001]

La distribución generada por esta función es almacenada en un archivo, y como último paso se han tomado las primeras 350 noticias de forma manual y se han colocado en un archivo diferente, definido así las noticias de entrenamiento (con 3,150) y de prueba (con 350).

La cantidad de noticias por sección del conjunto de entrenamiento se muestran en la Figura 5.16 y del conjunto de prueba en la Figura 5.17.

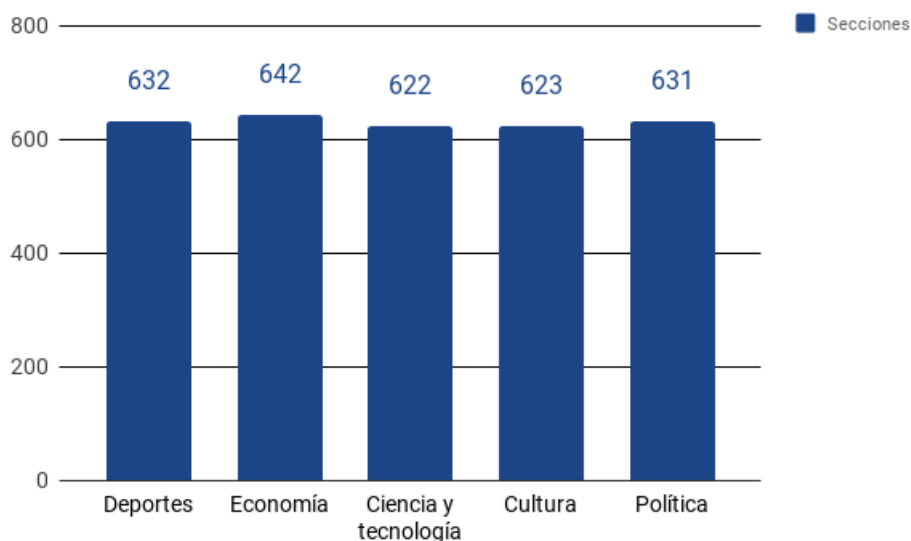


Figura 5.16: Corpus de entrenamiento

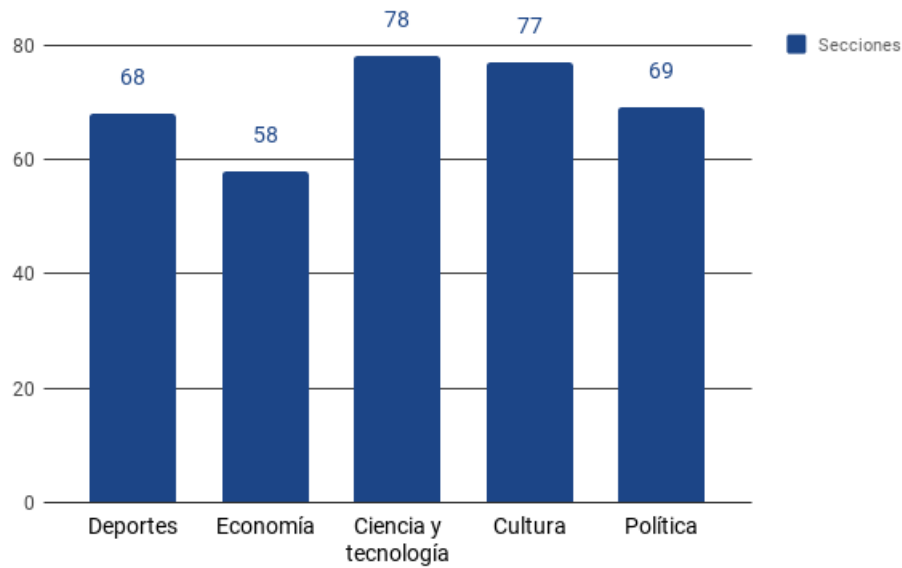


Figura 5.17: Corpus de prueba

5.2.2. Entrenamiento

Para crear un modelo clasificador usando aprendizaje supervisado (ver [Aprendizaje supervisado](#)), se debe construir dos conjuntos etiquetados: entrenamiento para el proceso de aprendizaje y otro de prueba, para medir su precisión. En la sección anterior estos grupos se han formado con noticias de 5 secciones: **Deportes**, **Economía**, **Política**, **Cultura**, **Ciencia y tecnología**. Ambos conjuntos de datos serán usados por 4 algoritmos (seleccionados con base en el estado del arte ver [Estado del arte](#)), los cuales son:

- **Naive Bayes** (ver [Naive Bayes](#))
- **Regresión logística** (ver [Regresión logística](#))
- **Máquina de soporte vectorial** (ver [MSV](#))
- **Random Forest** (ver [Random Forest](#))

El proceso de entrenamiento consta de 5 pasos los cuales se muestran en la Figura 5.18.

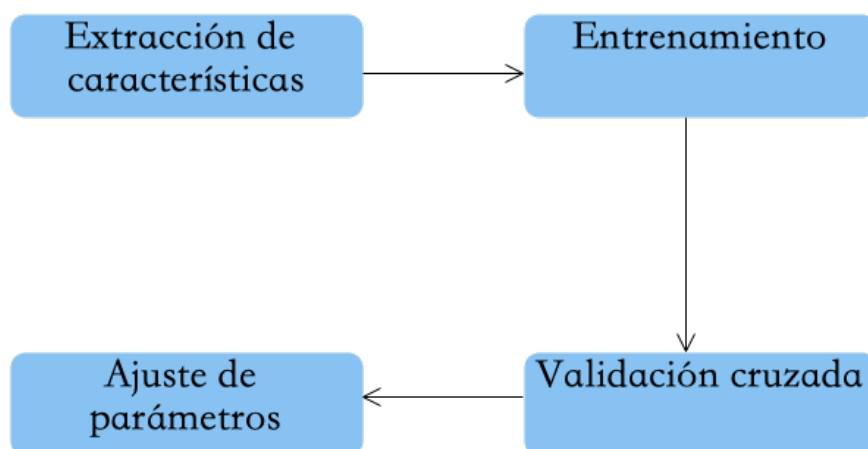


Figura 5.18: Etapas de entrenamiento

El desarrollo se ha implementado en el lenguaje de programación **Python 3**, utilizando la biblioteca **scikit learn** quien permite crear instancias de los algoritmos mencionados.

Como primer paso del entrenamiento, el corpus debe ser representado mediante conjuntos de vectores numéricos, esta es llamada representación vectorial (ver [Representación del texto](#)). Para lograr este objetivo, del corpus se deben seleccionar características que definan los elementos del vector numérico.

5.2.2.1. Extracción de características

La extracción de características cuenta con dos tareas importantes: extraer el vocabulario y crear un vector de características.

Cada clase de noticias contiene un conjunto de palabras que son comunes en su ámbito, al analizar el léxico usado se observa los tecnicismos usados, por ejemplo en la sección deportes se ocupa, fútbol, jugador, ganador; en política, presidente, corrupción, PRI; en ciencia y tecnología, investigación, descubrimiento, publicación y así sucesivamente, por lo tanto estos vocablos pueden ser definidos como características que identifican a una sección. En este sentido la extracción de características es el proceso de tomar las pala-

bras de las noticias para formar un vocabulario.

Para ejemplificar esta tarea observe el Cuadro 5.2.8 el cual es un corpus de 4 oraciones. Una vez realizado el proceso de extracción de características se obtiene el vocabulario el cual el mostrado en el Cuadro 5.2.9.

Cuadro 5.2.8: Corpus

<i>Este</i>	<i>es</i>	<i>la</i>	<i>primera</i>	<i>noticia</i>	<i>noticia</i>	
<i>Esta</i>	<i>noticia</i>	<i>es</i>	<i>la</i>	<i>segunda</i>	<i>noticia</i>	<i>???</i>
<i>Y</i>	<i>este</i>	<i>es</i>	<i>la</i>	<i>tercera</i>		
<i>Es</i>	<i>este</i>	<i>la</i>	<i>primera</i>	<i>noticia</i>		<i>????</i>

Cuadro 5.2.9: Selección de características

[*es esta este la noticia primera segunda tercera y ?*]

Después de extraer el vocabulario, se crea un espacio vectorial por cada noticia donde cada elemento del vector representa la presencia o ausencia de una característica (palabra). Cabe mencionar que las características son extraídas de 2 formas, binario (donde 1 representa la presencia de la característica y 0 la ausencia) y por frecuencia (donde se cuenta el número de veces que cada característica aparece). Continuando con el ejemplo del Cuadro 5.2.8 se extraen las características por frecuencia y el resultado se muestra en el Cuadro 5.2.10, mientras que el Cuadro 5.2.11 muestra las características extraídas de forma binaria.

Para el desarrollo de esta etapa se ha utilizado la biblioteca **CountVectorizer** quien permite generar la selección y extracción de características, además esta es la representación vectorial (de cada noticia) que los algoritmos de clasificación pueden entender y por ende ser entrenados. Cabe destacar que la representación vectorial usada en este trabajo ha sido de forma binaria.

Cuadro 5.2.10: Representación por frecuencia

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 2 & 0 & 1 & 0 & 0 & 3 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 4 \end{bmatrix}$$

Cuadro 5.2.11: Representación binaria

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

5.2.2.2. Entrenamiento

El corpus contiene noticias de varias fuentes, en las cuales la redacción, coherencia, semántica varía, incluso en la edición de una misma noticia, por lo tanto en el entrenamiento se busca generalizar la clasificación de los artículos, analizando el texto como un conjunto de palabras, sin tomar en cuenta la semántica, esta técnica es llamada bolsa de palabras (ver [Bolsa de palabras](#)).

En este punto del trabajo las noticias están representadas en un espacio vectorial, y serán usadas en el proceso de entrenamiento de los algoritmos: **Naive Bayes**, **Regresión logística**, **Máquina de soporte vectorial**, **Random Forest**. Cada clasificador recibe como entrada un conjunto de vectores etiquetados y como salida se genera un modelo el cual predice la sección de nuevas noticias.

Para este trabajo se han definido las etiquetas como se muestra en la Tabla 5.5, esta correspondencia es de secciones de noticias a un número único.

Sección	Etiqueta
Deportes	0
Economía	1
Política	2
Cultura	3
Ciencia y tecnología	4

Tabla 5.5: Etiquetas de secciones

El desarrollo ha utilizado una instancia de cada algoritmo, para esto se incluye la biblioteca correspondiente de **scikitlearn**, las cuales se muestran en la Tabla 5.6, la entrada al clasificador es el conjunto de características y las etiquetas correspondientes, esto regresa como resultado un modelo que es capaz de predecir la sección de noticias, sin que estas estén etiquetadas.

Sección	Etiqueta
Naive Bayes	MultinomialNB
Máquina de soporte vectorial	SVC
Regresión logística	LogisticRegression
Random Forest	RandomForestClassifier

Tabla 5.6: Biblioteca de algoritmo

5.2.2.3. Validación cruzada

En el proceso de entrenamiento, los clasificadores reciben un conjunto noticias para ser entrenados y otro para realizar pruebas, sin embargo la selección de los artículos puede ser manipulada para obtener un resultado a conveniencia, siendo esto una mala práctica, por esta razón y en con el objetivo de obtener resultados mas robustos se ha implementado un técnica llamada Validación cruzada (ver [Validación cruzada](#)).

Este método consiste en tres pasos: el primero es dividir el corpus en entrenamiento y prueba (este conjunto es llamado pliegue); después se calcula la exactitud de la prueba y es almacenado; como último etapa los dos primeros paso son repetidos n veces y para terminar se calcula el promedio de la exactitud. En términos generales este promedio nos brinda mayor confianza

en el resultado del entrenamiento de cada clasificador.

5.2.2.4. Ajuste de parámetros

Como se ha visto, los resultados de la clasificación son medidos por la cantidad de noticias correctas clasificadas, no obstante estos resultados pueden incrementar o decrementar con base a los parámetros ingresados a cada algoritmo.

Esta etapa consiste en observar el mejor resultado en la clasificación con los diferentes algoritmos, variando los parámetros y aplicar validación cruzada para medir la precisión. Para ejemplificar este tarea la Figura 5.19 muestra la variación del parámetro **alpha** en el algoritmo **Naive bayes** (este parámetro será explicado mas adelante) tomando los valores: 0.5, 1.0, 1.5 y 2.0, aplicando 2 pliegues en la validación cruzada.

```
[CV] alpha=0.5 .....
[CV] ..... alpha=0.5, score=0.853, total= 0.0s
[Parallel(n_jobs=1)]: Done 1 out of 1 | elapsed: 0.0s remaining:
[CV] alpha=0.5 .....
[CV] ..... alpha=0.5, score=0.835, total= 0.0s
[Parallel(n_jobs=1)]: Done 2 out of 2 | elapsed: 0.0s remaining:
[CV] alpha=1 .....
[CV] ..... alpha=1, score=0.845, total= 0.0s
[CV] alpha=1 .....
[CV] ..... alpha=1, score=0.835, total= 0.0s
[CV] alpha=1.5 .....
[CV] ..... alpha=1.5, score=0.841, total= 0.0s
[CV] alpha=1.5 .....
[CV] ..... alpha=1.5, score=0.834, total= 0.0s
[CV] alpha=2 .....
[CV] ..... alpha=2, score=0.836, total= 0.0s
[CV] alpha=2 .....
[CV] ..... alpha=2, score=0.827, total= 0.0s
```

Figura 5.19: Corpus de entrenamiento

Se observa que por cada parámetro implementado en la validación cruzada se muestra el resultado **score**, el cual es el promedio de la precisión de dicha prueba, se puede observar que el mejor resultado ha sido hecha con el

$\alpha = 0,5$, con un 85 % de precisión.

A continuación se explican los parámetros de cada clasificador:

Naive Bayes

El parámetro en el cual se varía en este algoritmo, es un escalar llamado **Alpha** (α). Este es un valor numérico asignado a cada palabra en el corpus con respecto a la frecuencia de aparición en un clase. El valor α evita que la importancia de una palabra se haga cero, **Alpha** es variado en los valores $[0,5, 1,0, 1,5, 2]$.

Regresión logística

Este algoritmo está basado en una regresión lineal y el calculo de probabilidades como se explica en el capitulo 3 (ver [Regresión logística](#)). Existen 2 parámetros importantes en este clasificador, optimizando la función l_1 y minimizando el costo de la función l_2 , en el cual existe un escalar C , donde: para valores pequeños de este número, los valores de los pesos w (la importancia de cada palabra) se decrementa, teniendo así un modelo muy simple (se pierde información), de lo contrario para valores grandes de C la complejidad del modelo aumenta pero se incrementa el ruido.

Los parámetros de este algoritmo son: l_1 y l_2 que es la forma de entrenar el algoritmo; C que es el equilibrio entre la simplicidad del algoritmo y la tolerancia de ruido, el cual toma los valores $: [1e - 6, 1e - 05, 1e - 04, 1e - 03, 1e - 02, 1e - 01]$.

Random forest

Este algoritmo está basado en la construcción de conjuntos de arboles de decisión como se explica en el capitulo 3 (ver [Random Forest](#)), donde se tienen que controlar la cantidad de árboles a crear y la profundidad. Cabe señalar que a mas profundidad las características son separadas con mas homogeneidad, sin embargo este es un problema, si el árbol creado es completamente homogéneo sufrirá de sobreentrenamiento, es decir hará un buen trabajo clasificando el conjunto de entrenamiento pero lo hará mal con el conjunto de prueba.

Los parámetros utilizados en ese algoritmo son; **n_estimators** es la cantidad de árboles creados, el cual esta en el rango de : [50, 100, 500, 1000]; **max_depth** es la profundidad del árbol y ha sido establecida en los rangos: [50, 100, 500, 1000].

Máquina de soporte vectorial

Este algoritmo clasifica los datos con un margen que equidiste de los tipos de clases (secciones), sin embargo en el caso de que los datos no son separables en la dimensión inicial, se busca encontrar la solución con hiperplanos en dimensiones superiores (ver [MSV](#)).

Para dividir los datos de forma correcta se busca usar funciones de división la cual es denominada el kernel de la función. En este trabajo se han probado 3 tipos de kernel:

- **Lineal:** $\langle x, x' \rangle$ (Figura 5.20a)
- **Polinomial:** $(\gamma \langle x, x' \rangle + r)^d$ (Figura Figura 5.20b)
- **Radial:** $\exp(-\gamma \|x - x'\|^2)$ (Figura 5.21)

Los parámetros de este clasificador son definidos por el kernel utilizado, para el kernel **polinomial** y **Radial** se usó el parámetro γ con los valores: $[1e - 4, 1e - 5, 1e - 6]$. Además como se explicó en **regresión logística** se varía el escalar C , el cual ajusta la variación y el bias de los datos, a mayor bias se sobreentrena el modelo, es decir que el modelo se ajusta demasiado a los datos de entrenamiento, a mayor varianza los datos de entrenamiento se ajustan menos, en ambos casos el desempeño con nuevos datos se vuelve deficiente. Por esta razón se tiene que encontrar un equilibrio en estos datos.

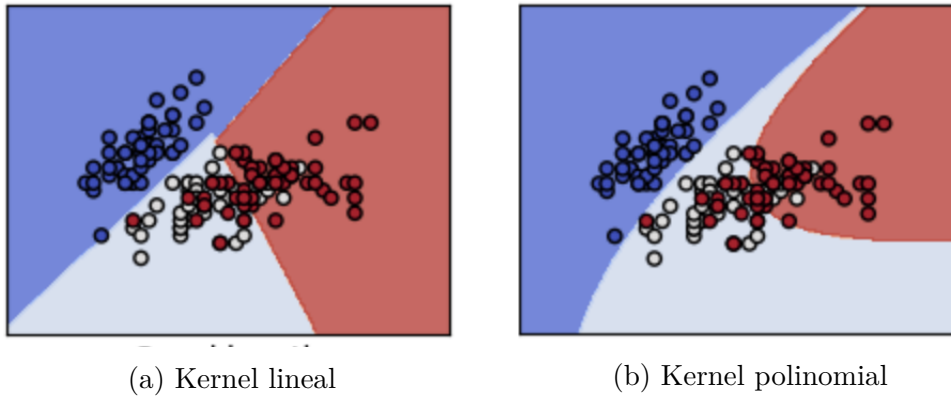


Figura 5.20: Kernel de la función (Pedregosa et al., 2011)

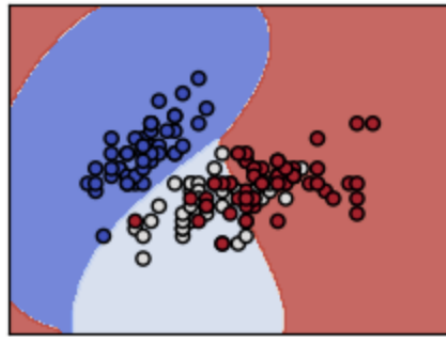


Figura 5.21: Kernel radial (Pedregosa et al., 2011)

Ahora se han explicado todos los parámetros de los algoritmos. Para las pruebas de este trabajo terminal han implementado 5 pliegues en la validación cruzada y se ha usado el corpus con 3150 noticias. Los resultados son visualizados en las siguientes tablas, donde la primera columna **Parámetros** contiene los parámetros variados de cada prueba, las columnas siguientes **P1**, **P2**, **P3**, **P4** y **P5** muestran el **exactitud** obtenido en cada pliegue, la columna **Promedio** muestra la suma de los **exactitud** obtenidos dividido entre 5 (número de pliegues) y la última columna **Rank** muestra un número entero el cual indica el *Ranking* de cada prueba, es decir la prueba con rank 1 es el mejor resultado, la prueba con rank 2 es el segundo mejor y así sucesivamente.

Parámetros	P 1	P 2	P 3	P 4	P 5	Promedio	Rank
alpha: 0.5	0.8610	0.8576	0.8458	0.8487	0.8424	0.8511	1
alpha: 1	0.8531	0.8497	0.8426	0.8408	0.8360	0.8444	2
alpha: 1.5	0.8531	0.8434	0.8410	0.8392	0.8312	0.8416	3
alpha: 2	0.8531	0.8418	0.8410	0.8392	0.8296	0.8409	4

Tabla 5.7: Naive bayes

Parámetros	P 1	P 2	P 3	P 4	P 5	Promedio	Rank
C: 0.1, penalty: l2, solver: liblinear	0.8768	0.8655	0.8521	0.8710	0.8758	0.8682	1
C: 0.01, penalty: l2, solver: liblinear	0.8705	0.8655	0.8474	0.8742	0.8615	0.8638	2
C: 0.1, penalty: l1, solver: liblinear	0.8515	0.8544	0.8442	0.8424	0.8376	0.8460	3
C: 0.001, penalty: l2, solver: liblinear	0.8436	0.8560	0.8299	0.8392	0.8169	0.8371	4
C: 0.0001, penalty: l2, solver: liblinear	0.7836	0.7832	0.7806	0.7739	0.7723	0.7787	5
C: 0.01, penalty: l1, solver: liblinear	0.6477	0.6297	0.5946	0.6099	0.6099	0.6184	6
C: 1e-05, penalty: l2, solver: liblinear	0.6051	0.6044	0.6057	0.5924	0.5796	0.5974	7
C: 1e-06, penalty: l2, solver: liblinear	0.5261	0.5316	0.5564	0.5223	0.5048	0.5282	8
C: 1e-06, penalty: l1, solver: liblinear	0.2006	0.2009	0.2003	0.2006	0.2006	0.2006	9
C: 1e-05, penalty: l1, solver: liblinear	0.2006	0.2009	0.2003	0.2006	0.2006	0.2006	9
C: 0.0001, penalty: l1, solver: liblinear	0.2006	0.2009	0.2003	0.2006	0.2006	0.2006	9
C: 0.001, penalty: l1, solver: liblinear	0.2006	0.2009	0.2003	0.2006	0.2006	0.2006	9

Tabla 5.8: Regresión logística

Parámetros	P 1	P 2	P 3	P 4	P 5	Promedio	Rank
max_depth: 50, n_estimators: 1000	0.8720	0.8592	0.8537	0.8631	0.8615	0.8619	1
max_depth: 1000, n_estimators: 1000	0.8689	0.8639	0.8585	0.8583	0.8583	0.8616	2
max_depth: 100, n_estimators: 1000	0.8752	0.8592	0.8569	0.8599	0.8551	0.8613	3
max_depth: 100, n_estimators: 500	0.8784	0.8608	0.8601	0.8535	0.8519	0.8609	4
max_depth: 1000, n_estimators: 500	0.8720	0.8528	0.8585	0.8551	0.8599	0.8597	5
max_depth: 50, n_estimators: 500	0.8705	0.8544	0.8601	0.8535	0.8599	0.8597	6
max_depth: 500, n_estimators: 1000	0.8720	0.8655	0.8506	0.8487	0.8583	0.8590	7
max_depth: 500, n_estimators: 500	0.8689	0.8544	0.8553	0.8519	0.8615	0.8584	8
max_depth: 500, n_estimators: 100	0.8610	0.8608	0.8506	0.8487	0.8662	0.8575	9
max_depth: 50, n_estimators: 100	0.8641	0.8528	0.8601	0.8503	0.8455	0.8546	10
max_depth: 1000, n_estimators: 100	0.8594	0.8528	0.8474	0.8392	0.8439	0.8485	11
max_depth: 100, n_estimators: 100	0.8657	0.8560	0.8394	0.8471	0.8280	0.8473	12
max_depth: 50, n_estimators: 50	0.8626	0.8402	0.8267	0.8583	0.8471	0.8470	13
max_depth: 100, n_estimators: 50	0.8420	0.8402	0.8283	0.8503	0.8535	0.8429	14
max_depth: 500, n_estimators: 50	0.8531	0.8418	0.8331	0.8424	0.8312	0.8403	15
max_depth: 1000, n_estimators: 50	0.8578	0.8212	0.8315	0.8392	0.8280	0.8355	16

Tabla 5.9: Random Forest

Parámetros	P 1	P 2	P 3	P 4	P 5	Promedio	Rank
C: 100, gamma: 0.0001, kernel: rbf	0.8736	0.8639	0.8537	0.8726	0.8822	0.8692	1
C: 1000, gamma: 1e-05, kernel: rbf	0.8752	0.8623	0.8537	0.8710	0.8822	0.8689	2
C: 1000, gamma: 0.0001, kernel: rbf	0.8689	0.8608	0.8506	0.8694	0.8790	0.8657	3
C: 1, gamma: 0.0001, kernel: linear	0.8657	0.8592	0.8410	0.8710	0.8742	0.8622	4
C: 1, gamma: 1e-05, kernel: linear	0.8657	0.8592	0.8410	0.8710	0.8742	0.8622	4
C: 1, gamma: 1e-06, kernel: linear	0.8657	0.8592	0.8410	0.8710	0.8742	0.8622	4
C: 10, gamma: 0.0001, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 10, gamma: 1e-05, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 10, gamma: 1e-06, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 100, gamma: 0.0001, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 100, gamma: 1e-05, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 100, gamma: 1e-06, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 1000, gamma: 0.0001, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 1000, gamma: 1e-05, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 1000, gamma: 1e-06, kernel: linear	0.8657	0.8592	0.8394	0.8710	0.8742	0.8619	5
C: 10, gamma: 0.0001, kernel: rbf	0.8641	0.8576	0.8490	0.8678	0.8503	0.8578	6
C: 100, gamma: 1e-05, kernel: rbf	0.8641	0.8576	0.8490	0.8662	0.8519	0.8578	6
C: 1000, gamma: 1e-06, kernel: rbf	0.8641	0.8576	0.8490	0.8662	0.8519	0.8578	6
C: 100, gamma: 1e-06, kernel: rbf	0.6524	0.6772	0.6073	0.6449	0.6608	0.6485	7
C: 10, gamma: 1e-05, kernel: rbf	0.6493	0.6741	0.6073	0.6401	0.6561	0.6454	8
C: 1, gamma: 0.0001, kernel: rbf	0.6193	0.6440	0.5946	0.6115	0.6306	0.6200	9
C: 1, gamma: 0.0001, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 1, gamma: 1e-05, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 1, gamma: 1e-05, kernel: rbf	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 1, gamma: 1e-06, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 1, gamma: 1e-06, kernel: rbf	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 10, gamma: 0.0001, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 10, gamma: 1e-05, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 10, gamma: 1e-06, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 10, gamma: 1e-06, kernel: rbf	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 100, gamma: 0.0001, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 100, gamma: 1e-05, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 100, gamma: 1e-06, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 1000, gamma: 0.0001, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 1000, gamma: 1e-05, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10
C: 1000, gamma: 1e-06, kernel: poly	0.2038	0.2041	0.2035	0.2038	0.2038	0.2038	10

Tabla 5.10: Máquina de soporte vectorial

5.2.3. Selección

Los resultados de la sección anterior ha mostrado que no existe una brecha muy grande en el **exactitud** obtenido de los mejores parámetros, como se muestra en la siguiente Tabla (5.11).

Algoritmo	Precisión	Ranking
MSV	0.8692	1
Regresión Logística	0.8619	2
Random Forest	0.8682	3
Naive Bayes	0.8511	4

Tabla 5.11: Precisión de los mejores parámetros

El mejor resultado se ha conseguido con el algoritmo **Máquina de soporte vectorial** con los parámetros ; $C=100$, $gamma = 0,0001$ y $kernel= rbf$ (radial) obteniendo 0.8619 de **exactitud**, por esta razón se ha elegido como el clasificador final de este trabajo terminal.

5.2.4. Pruebas

Con base en el algoritmo elegido y los parámetros que han conseguido obtener el mejor resultado, se ha hecho una prueba final, la cual consiste en clasificar el corpus de prueba con 350 noticias, creado al final de la sección de pre-procesamiento (ver [División del corpus](#)), para calcular la matriz de confusión y obtener las métricas de evaluación. La tabla 5.12 muestra el número de noticias por sección contenidas en el corpus.

Sección	Número de noticias
Deportes	68
Economía	58
Política	69
Cultura	77
Ciencia y Tecnología	78

Tabla 5.12: Número de noticias

La matriz de confusión obtenida se muestra en la tabla 5.13, donde las filas representan la clasificación real de la noticia y las columnas lo predicho por el clasificador. Por ejemplo se observa que en la sección **Deportes** de 68 noticias etiquetadas, 65 fueron clasificadas correctamente, y 3 fueron mal clasificadas en la sección **ciencia y tecnología** (1 noticia), **economía** (1 noticia) y **política** (1 noticia). En la sección de ciencia y tecnología de 78

noticias solo 67 fueron clasificadas de forma acertada, y 11 fueron clasificadas en las distintas secciones: 1 en **Deportes**, 5 en **Economía**, 3 en **Política** y 2 en **Cultura**.

Sección	Deportes	Economía	Política	Cultura	Cienci y T
Deportes	65	1	1	0	1
Economía	1	51	4	1	1
Política	2	5	59	3	0
Cultura	1	1	1	70	4
Ciencia y T	1	5	3	2	67

Tabla 5.13: Matriz de confusión

Con base en la matriz de confusión se ha calculado **Recall**, **Fmeasure**, **Precision** (ver [Métricas de evaluación](#)). Las métricas son obtenidas por cada sección, como se muestra en la tabla 5.14.

Sección	Precision	Recall	F-measure
Deportes	0.93	0.96	0.94
Economía	0.81	0.88	0.84
Política	0.87	0.86	0.86
Cultura	0.92	0.91	0.92
Ciencia y Tecnología	0.92	0.86	0.89

Tabla 5.14: Metricas de evaluación

Como se puede observar en la tabla 5.14 las sección con mejor **Fmeasure** (este valor sostiene una relación entre **Precision** y **Recall**) es **Deportes** obteniendo 0.94 en la clasificación. Este resultado se puede explicar con base en el vocabulario manejado en la categoría, ya que es muy específico, por ejemplo se usan palabras como: gol, jugador, resultado, anotación. Por lo contrario el **Fmeasure** mas bajo lo obtuvo **Economía** con 0.84 (el cual no es un mal resultado), las palabras manejadas por esta sección pueden ser encontradas en otras categorías, generando así confusión en la clasificación.

Finalmente, considerando la exactitud de de todas las secciones, el modelo entrenado obtuvo una exactitud promedio de 0.89. Este resultado es muy

bueno considerando que los trabajos descritos en el estado del arte (ver [Estado del arte](#)) obtienen una exactitud entre 0.80 y 0.90

5.2.5. Persistencia

Como última etapa de la clasificación, el modelo **Máquina de soporte vectorial** debe ser almacenado así como las características del corpus. Para esta tarea se ha utilizado el corpus completo de 3500 noticias, del cual se extraen las características, se entrena el modelo y estos conjuntos son almacenados en un archivo ocupando la biblioteca **pickle**, de esta forma en la siguiente sección el modelo es utilizado como parte de la **Aplicación web**.

5.3. Aplicación web

Como se estableció en el objetivo de este trabajo, se ha creado una aplicación para recolectar y clasificar noticias, las cuales son mostrados en un entorno web. En esta sección se explica el proceso de funcionamiento de la aplicación web.

Para el desarrollo de esta herramienta se ha utilizado **Java** para el procesamiento de los datos y para diseñar la vista se ha utilizado: **HTML**, **CSS** y **Java server faces**, además para conectar el proceso de recolección y clasificación (escritos en **Python 3**) se ha hecho uso de **subprocesos**, los cuales son procesos que corren en segundo plano.

La aplicación está diseñada en 4 etapas, las cuales son mostradas en la Figura 5.22, donde la etapa **Selección** describe las secciones disponibles para obtener noticias; la etapa **Recolectar** muestra la integración de los *crawlers* en la herramienta; la etapa **Clasificar** hace uso del modelo clasificador desarrollado en la sección anterior y para concluir, la etapa **Mostrar resultados** describe la forma en que los datos son presentados al usuario. Es importante mencionar que las etapas en color azul cielo siempre son ejecutadas, sin embargo las de color gris no siempre son llevadas a cabo. A continuación se explica a detalle cada una de las tareas.

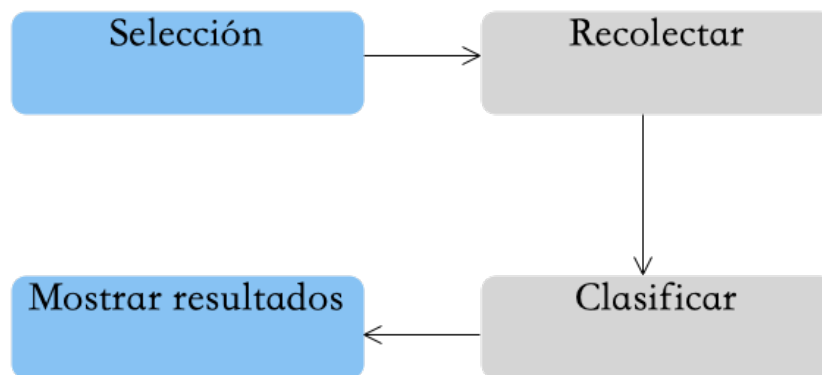


Figura 5.22: Etapas de la aplicación web

5.3.1. Selección

Esta etapa permite elegir al usuario la sección de noticias a buscar. La aplicación comienza en una pantalla inicial, donde se muestra un menú con las opciones **Inicio**, **Cultura**, **Deportes**, **Economía**, **Política**, **Ciencia y Tecnología**, estas categorías (excluyendo **Inicio**) son las secciones permitidas para recolectar noticias, como se muestra en la Figura 5.23. Cabe mencionar que la opción **Inicio**, permite regresar a la pantalla principal.



Figura 5.23: Pantalla de Inicio

Después de elegir una sección, se muestra el mensaje **En proceso de recolección y clasificación**, como se visualiza en la Figura 5.24, el cual informa que las etapas **Recolectar** y **Clasificar** están en proceso, además cuando se muestra este mensaje no se puede seleccionar otra secciones hasta que el proceso concluya. Cabe destacar que el proceso continua de forma normal si se cumplen las siguientes condiciones:

1. **Primera vez:** Esta condición hace referencia a la etapa **selección**, es decir cuando es la primera vez que se ha selecciona una sección, se debe proceder con las siguientes etapas
2. **Límite de periodo:** Esta condición define 4 horas como el periodo de recolección, es decir cuando se ha solicitado mostrar las noticias de una

sección y ha transcurrido 4 horas desde la última petición (Ver regla de negocio [RN8 Periodo de actualización](#)), se debe proceder con las siguientes etapas

Como consecuencia de no cumplirse estas condiciones, las etapas **Recolectar** y **Clasificar** no son iniciadas, debido a que los artículos con su clasificación correspondiente se encuentran almacenados en el sistema, por esta razón se procede directamente con la etapa **Mostrar resultado**.



Figura 5.24: Mensaje de espera

5.3.2. Recolectar

Esta etapa genera un subproceso el cual activa un script (desarrollado en **Python 3**) el cual activa la recolección de noticias en los sitio web definidos previamente (ver [Sitios web](#)), la información que se obtiene de cada artículo es la siguiente:

- URL de la noticia
- Título
- Fecha
- Autor

- **Descripción** (Existen sitios web, donde las noticias no cuentan con una descripción)
- **Noticia**

La extracción de las noticias se hace en la página principal de los sitios web. Cabe destacar que en el proceso de recolección se valida que las noticias contengan al menos 180 palabras (en la redacción de **Noticia**), de lo contrario no se extrae. Además se ha definido un tiempo máximo de espera en esta etapa, el cual es de 84 segundos, después de concluir el periodo y haber recolectado la información, se procede con la etapa de **Clasificación**, de lo contrario si no se recolectó ninguna noticia se muestra el mensaje **Se ha agotado el tiempo de espera, no se han encontrado noticias, intentar mas tarde** (como se visualiza en la Figura 5.25) y se detiene el proceso.



Figura 5.25: Mensaje de error en la recolección

5.3.3. Clasificar

Después de recolectar las noticias se inicia la etapa de clasificación, el cual esta conformado por 5 tareas (como se muestra en la Figura ??). Esta etapa genera un subproceso para ejecutar un script desarrollado en **Python 3**, el cual está encargado de llevar acabo cada tarea de esta sección. Cabe destacar que el

proceso de clasificación solo utiliza el contenido de la noticia, los demás datos no son necesarios en esta etapa. A continuación se explica cada subtarea.

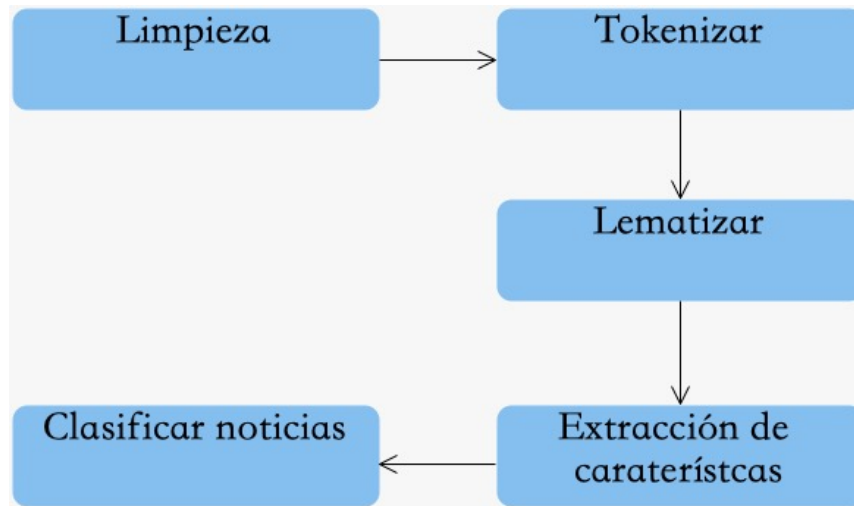


Figura 5.26: Proceso de clasificación

1. **Limpieza:** En la sección de **Entrenamiento** el proceso de limpieza (ver [Limpieza](#)) consiste en eliminar texto que no brinda información útil como, hipertexto (ver [HTML](#)), símbolos especiales (como # † √), *emojis* (como 🤔 😬 🍷), sin embargo en esta etapa no es necesario hacer esta limpieza, debido a que el vocabulario definido no contiene estos datos, por esta razón estos símbolos son ignorados en la extracción de características. Cabe señalar que lo único que es eliminado son los saltos de línea
2. **Tokenizar:** Consiste en separar el texto en sus elementos mínimos por un espacio (ver [Tokenizar](#))
3. **Lematizar:** Proceso que reduce cada una de las palabras tokenizadas en lemas (ver [Lematizar](#))
4. **Extracción de características:** Se extrae palabras (características) con base al vocabulario definido al final de la etapa de entrenamiento (ver [Persistencia](#)), para crear un espacio vectorial por cada noticia (esta es la representación que el algoritmo entiende). Cabe destacar que

las características son extraídas de forma binaria (ver [Extracción de características](#))

5. **Clasificar:** Al final de la sección **Entrenamiento** se almacenó el modelo clasificador (ver [Persistencia](#)) el cual esta basado en el algoritmo **Maquinas de soporte vectorial** (ver [MSV](#)). Esta tarea utiliza el clasificador el cual recibe como entrada los vectores de características (los cuales representan el contenido de cada artículo) y como salida brinda la clasificación del conjunto de noticias, y son almacenadas por sección en un archivo **CSV**

5.3.4. Mostrar resultados

Esta etapa consiste en presentar el resultado del proceso de clasificación en la herramienta. Cuando este proceso ha concluido se muestra el mensaje **Noticias listas para ser mostradas**, donde el usuario tiene la opción de elegir visualizar las noticias o cancelar el flujo del sistema, como se muestra en la Figura 5.27.

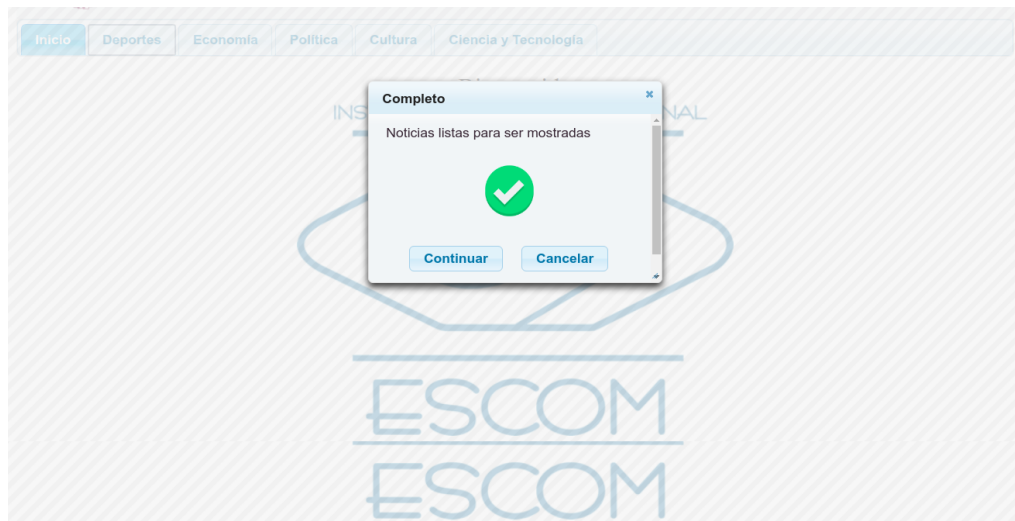


Figura 5.27: Mensaje que se muestra una vez clasificadas las noticias

Si el usuario ha presionado la opción **Cancelar**, el proceso concluye y la herramienta muestra la Pantalla de Inicio (ver 5.23), de lo contrario si se ha dado clic en **continuar**, se lleva acabo el proceso mostrado en la Figura 5.28,

las tareas en color azul cielo siempre son llevadas acabo y las de color gris son opcionales. A continuación estas etapas son descritas.

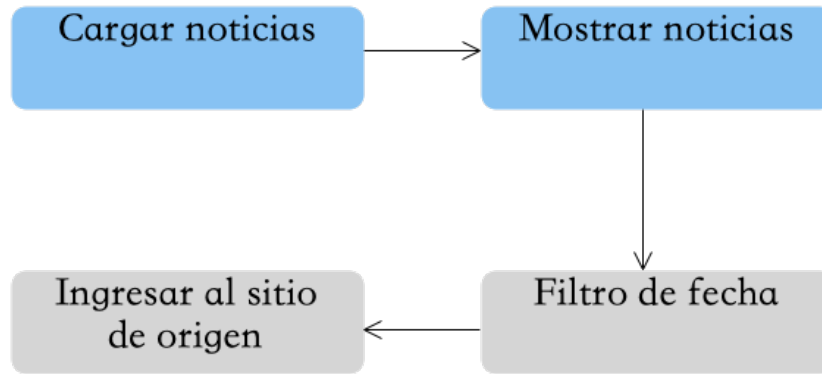


Figura 5.28: Funcionalidad de la aplicación

1. **Cargar noticias:** En el proceso de clasificación se crea un archivo **CSV** por cada sección con las noticias correspondientes, en esta etapa se obtiene los artículos de la sección elegida por el usuario
2. **Mostrar noticias:** En la pantalla principal, las noticias se muestran por fecha de publicación, los datos mostrados por cada noticia son: **título de la noticia**, **resumen de la noticia**(de contar con ello), **autor** y finalmente la **fecha de publicación**. Como se muestra en la Figura 5.29
3. **Filtro fecha:** En la sección de las noticias, se muestra un menú con las opciones: **Hoy**, **Ayer**, **Dos días** y **Tres días o mas**. Este es una herramienta que permite filtrar los artículos por fecha de publicación. Por ejemplo, si el usuario desea visualizar noticias de un día anterior a la registrada por el sistema, se debe seleccionar la opción **Ayer**, en seguida la información de este periodo se muestra, como en la Figura 5.30. Es importante mencionar que las noticias mostradas por defecto son las del día de consulta (**Hoy**)
4. **Ingresar al sitio de origen:** Cada noticia mostrada contiene la **URL** al sitio de origen de la noticia, si el usuario desea leer el artículo completo se debe dar clic en la noticia y la aplicación redirige el buscador a la pagina fuente



Figura 5.29: Vista de las noticias recolectadas



Figura 5.30: Vista de las noticias recolectadas del día de ayer

Capítulo 6

Conclusión



6.1. Conclusiones

La cantidad de páginas electrónicas, para consultar noticias ha incrementado con el paso de los años, su información al igual que un diario tradicional se encuentra dividida en secciones para facilitar la consulta, sin embargo, la clasificación suele variar en cada portal, incluso teniendo el mismo contenido. Por esta razón, buscar artículos de interés se ha hecho una tarea complicada y laboriosa. Para abordar una solución con base en este problema, el presente trabajo ha desarrollado una aplicación web, el cual permite recopilar noticias de diferentes fuentes y mediante un algoritmo son clasificadas automáticamente en 5 secciones diferentes.

Como primer punto abordado en el desarrollo de este trabajo, se debe mencionar que uno de los objetivos particulares, el cual es afinar el clasificador

del trabajo terminal 2017-A042, sin embargo, debido a los diferentes enfoques de clasificación se decidió generar un nuevo modelo y corpus.

Se **recolecto** un corpus de 3,500 noticias, con el fin de tener información suficiente para entrenar y probar 4 algoritmos de clasificación de la biblioteca **scikit-learn** los cuales son: **Naive Bayes**, **Regresión logística**, **Random Forest** y **Maquina de soporte vectorial**. Además para medir la eficiencia de los algoritmos se ha hecho uso de la técnica validación cruzada, el cual consiste en dividir un corpus en conjunto de entrenamiento y prueba, calcular la exactitud de cada prueba y obtener el promedio total. **Maquina de soporte vectorial** ha tenido el mejor resultado **con una exactitud de 0.869**, por esta razón el modelo clasificador se ha entrenado con este algoritmo.

Para recolectar las noticias se ha utilizado la **impelentación** de la **librería scrapy**, **el** cual permite crear arañas que extraen los elementos de las noticias (**título**, **URL**, **Contenido**, etc) de 7 sitios web. Cabe destacar que el desarrollo de ambas herramientas (modelo clasificador y recolector) se ha hecho en el lenguaje **python 3**.

Como segundo aspecto a abarcar, la implementación de la aplicación web tiene dos partes fundamentales: la vista y el procesamiento de los datos. Para esta última parte se ha usado **Java** como lenguaje de programación, además para conectar esta etapa con la recolección y clasificación se han usado subprocesos, los cuales son iniciados en la aplicación. En la primera etapa (recolección) se obtienen noticias de los sitios web, al concluir esta tarea la segunda etapa es iniciada (clasificación), en la cual se hace un procesamiento al contenido de la noticia, para pasar el contenido en lenguaje natural a un vector de características ocupado para la clasificación de las noticias, al concluir esta tarea, se obtiene el resultado y es presentado en la vista. El diseño del entorno web se ha hecho en **HTML**, **CSS** y **Java server faces**, este muestra al usuario las noticias divididas en las 5 secciones definidas. Cabe mencionar que el usuario tiene la opción de filtrar las noticias por fecha en las siguientes categorías: hoy, ayer, dos días y tres días o mas.

Como conclusión se puede decir que el objetivo principal se ha alcanzado, con buenos resultados en la clasificación.

6.2. Trabajo a futuro

Como trabajo a futuro se propone crear un servicio web, el cual pueda ser utilizado por el usuario con el fin de que sea más rápido el proceso de recolección y clasificación. Aunado a esto se propone agregar más sitios de los cuales puedan ser recolectadas noticias para su clasificación.

Adicional a lo mencionado, se propone crear una aplicación móvil que permita la visualización de noticias de distintos sitios web.



Bibliografía

- Aggarwal, C. C. (2018). *Machine Learning For Text*. Springer, primera edition.
- Alexey Grigorev, J. L. R. and Reese, R. M. (2017). *Java: Data Science Made Easy*. "Packt Publishing".
- Alfaro, E., Martínez, M. G., and Rubio, N. G. (2003). Una revisión de los métodos de agregación de clasificadores. In *Anales de economía aplicada 2003*, page 161. Asociación Española de Economía Aplicada, ASEPELT.
- Asyárie, A. D. and Pribadi, A. W. (2009). Automatic news articles classification in indonesian language by using naive bayes classifier method. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, pages 658–662, New York, NY, USA. ACM.
- B. Bracewell, D., Yan, J., Ren, F., and Kuroiwa, S. (2009). Category classification and topic discovery of japanese and english news articles. *Electr. Notes Theor. Comput. Sci.*, 225:51–65.
- Bellman, R. (1978). An introduction to artificial intelligence: Can computers think? pages 1–2, San Francisco. Boyd & Fraser.
- Berrar, D. (2018). *Encyclopedia of Bioinformatics and Computational Biology*. Elsevier.
- Blázquez, M. (2013). *Técnicas avanzadas de recuperación de información*. mblazquez, Madrid, 1aed edition.
- Bracewell, D. B., Ren, F., and Kuriowa, S. (2005). Multilingual single document keyword extraction for information retrieval. In *IEEE International*

Conference on Natural Language Processing and Knowledge Engineering, Wuhan, China, pages 517–522.

Breiman, L. (2001). *Machine learning*. Kluwer Academic Publishers.

Bruguera, E. (2019). Qué es un blog. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17821/5/XX0893006_01331-3.pdf.

CleverData (2019). Conceptos básicos de machine learning. <https://cleverdata.io/conceptos-basicos-machine-learning/>.

Economista, E. (2019). Ranking de medios nativos digitales. <https://www.eleconomista.com.mx/Ranking-de-Medios-Nativos-Digitales>.

Farias, G., Vergara, S., Fabregas, E., Hermosilla, G., Dormido-Canto, S., and Dormido, S. (2018). Clasificador de noticias usando autoencoders. In *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, pages 1–6. IEEE.

Fetherolf, H. B. J. W. R. M. (2016). *Real-World Machine Learning*. Manning Publications.

García, J., Ramírez, L., and Sánchez, M. (2018). Clasificación de noticias de diarios de circulación nacional mediante aprendizaje automático. Trabajo Terminal de ESCOM con número 2017-A04 (CDMX).

Google (2018). Googlebot. <https://www.humanlevel.com/diccionario-marketing-online/googlebot>.

Google (2019). Google cloud. <https://cloud.google.com/natural-language/?hl=Es-419>.

IBM (2017). Reconocimiento del lenguaje. <https://www.ibm.com/blogs/think/es-es/2017/05/16/watson-nlc-en-hogwarts/>.

Internaútica, I. (2018). Importancia de las noticias. <https://innovainternetmx.com/2014/12/importancia-de-las-noticias/>.

Kouzis-Loukas, D. (2016). *Learning Scrapy*. Packt Publishing Ltd, primera edición.

- Krol, K. (2019). *WordPress 5 Complete - Seventh Edition*. Packt Publishing Ltd.
- Manning, C., Raghavan, P., and Schütze, H. (2010). Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103.
- Marchal, B. (2001). *XML con ejemplos*. Number Sirsi) i9789702601630 QA76.76. H94. Pearson Educación.
- Marsland, S. (2014). *Machine Learning: An Algorithmic Perspective, Second Edition*. Chapman & Hall/CRC, 2nd edition.
- McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.
- McDaniel, A. (2011). *HTML5: Your visual blueprint for designing rich web pages and applications*, volume 37. John Wiley & Sons.
- McKinney, W. (2012). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. .o'Reilly Media, Inc."
- Mercer, D. (2006). *Drupal: Creating blogs, forums, portals, and community websites*. Packt Publishing Ltd.
- Mueller, J. P. and Massaron, L. (2016). *Machine learning for dummies*. John Wiley & Sons, 111 River St.
- Munzert, S., Rubba, C., Meißner, P., and Nyhuis, D. (2014a). *Automated data collection with R: A practical guide to web scraping and text mining*. John Wiley & Sons.
- Munzert, S., Rubba, C., Meissner, P., and Nyhuis, D. (2014b). *Automated data collection with R: A practical guide to web scraping and text mining*. John Wiley & Sons.
- Musciano, C. and Kennedy, B. (2002). *HTML & XHTML: The Definitive Guide: The Definitive Guide*. .o'Reilly Media, Inc."
- Pajares, G. and Santos, M. (2006). *Inteligencia artificial e ingeniería del conocimiento*. Alfaomega.

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Ramdass, D. and Seshasai, S. (2009). Document classification for newspaper articles. pages 1–11. Semantic scholar. <https://pdfs.semanticscholar.org/aa96/9114cf6e4d77c5bb3dd62a20bee3446f33ab.pdf>.
- Russell, S. and Norvig, P. (2009). Artificial intelligence: A modern approach. pages 28–29, Upper Saddle River, NJ, USA. Prentice Hall Press.
- Suárez, E. J. C. (2014). Tutorial sobre máquinas de vectores soporte (svm). *Tutorial sobre Máquinas de Vectores Soporte (SVM)*.
- Téllez-Valero, A., Montes, M., Fuentes, O., and Villaseñor-Pineda, L. (2019). Clasificación automática de textos de desastres naturales en México. In *Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)*.
- Téllez Valero, A., Montes, M., and Villaseñor-Pineda, L. (2009). Usando aprendizaje automático para extraer información de noticias de desastres naturales. *Computación y Sistemas*, 13:33–44.
- Vargiu, E. and Urru, M. (2013). Exploiting web scraping in a collaborative filtering-based approach to web advertising. *Artificial Intelligence Research*, 2:1–2.
- Velasco, M. S. (2002). La regresión logística . una aplicación a la demanda de estudios universitarios. 1:10.
- Visa, S., Ramsay, B., Ralescu, A. L., and Van Der Knaap, E. (2011). Confusion matrix-based feature selection. *MAICS*, 710:120–127.
- Wongso, R., Ariandy Luwinda, F., Christian Trisnajaya, B., Rusli, O., and , R. (2017). News article text classification in Indonesian language. *Procedia Computer Science*, 116:137–143.
- Yunta, L. R. (2006). La lematización en español: una aplicación para la recuperación de información (r. gómez díaz). *Revista española de Documentación Científica*, 29(1):175–176.