

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:

M. en C. JOEL OMAR JUÁREZ GAMBINO
Dra. CONSUELO VARINIA GARCÍA
MENDOZA



Ciudad de México, 8 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	4
2.1. Introducción	4
2.2. Trabajos nacionales	5
2.2.1. Clasificación de noticias de diarios	5
2.2.2. Desastres naturales en México	5
2.2.3. Extraer información de noticias de DN	6
2.3. Trabajos I	6
2.3.1. Clasificador de noticias usando autoencoders	6
2.3.2. Document classification for newspaper articles	7
2.3.3. Category classification and topic discovery	7
2.3.4. Automatic news articles classification in indonesian	9
2.3.5. News article text classification in indonesian language	10
2.4. Herramientas D	11
2.4.1. Cloud natural language	11
2.4.2. Googlebot	11
2.4.3. Watson natural language classifier	11
3. Marco teórico	13
3.1. Inteligencia Artificial	14
3.2. Procesamiento de LN	15
3.3. Aprendizaje Automático	15

3.4.	AA Para texto	16
3.4.1.	Representación del T	17
3.4.2.	Pre-procesamiento	17
3.4.3.	Extracción de texto	18
3.4.4.	Corpus	18
3.4.5.	Tokenización	18
3.4.6.	Lematización	19
3.4.7.	Frecuencia inversa del documento	19
3.5.	A Supervisado	19
3.5.1.	Regresión logística	20
3.5.2.	Naive bayes	21
3.5.3.	Maquina de soporte vectorial	22
3.5.4.	Random forest	23
3.6.	Métricas de evaluación	23
3.7.	Web Scraping	26
3.7.1.	Técnicas de web scraping	26
3.8.	Crawler	27
3.9.	Python	27
3.9.1.	Scrapy	28
3.10.	Internet	29
3.10.1.	World Wide Web	29
3.11.	HTML	30
3.12.	Sitios web	30
3.12.1.	Página web	30
3.12.2.	Blog	31
3.12.3.	Foro	31
4.	Análisis y diseño	32
4.1.	Actores y roles	32
4.2.	Requisitos f.	32
4.3.	Requisitos no f.	33
4.4.	Reglas de negocio	34
4.5.	Casos de uso	37
4.5.1.	Diagrama de casos de uso	37
4.5.2.	CU1 Recolectar noticias	38
4.5.3.	CU4 Mostrar resultados	41
4.5.4.	CU3 Pre-procesar noticias	44
4.6.	Mensajes	46

4.7. Pantallas	47
4.7.1. UI1 Página de inicio	47
4.7.2. UI2 Recolección y clasificación	48
4.7.3. UI3 Proceso concluido	49
4.7.4. UI4 Resultados de consulta	50
4.8. Diagrama de secuencia	54
5. Implementacion y Pruebas	55
5.1. Recolección	56
5.1.1. Selección de sitios web	56
5.1.2. Análisis de sitios web	58
5.1.3. Creación de recolector	58
5.1.4. Recolección de noticias	61
5.2. Entrenamiento	65
5.2.1. Preprocesamiento	65
5.2.2. Entrenamiento	72
5.2.3. Selección	80
5.2.4. Pruebas	80
5.2.5. Persistencia	80
5.3. Aplicación web	81
5.3.1. Tecnologías utilizadas	81
5.3.2. Proceso de recolección	81
5.3.3. Proceso de clasificación	82
5.3.4. Frontend	82
Bibliografía	84

Índice de figuras

3.1. Campo de estudio	13
3.2. Matriz de confusión	24
3.3. Etapas del proceso de <i>Web scraping</i>	27
4.1. Diagrama de casos de uso	37
4.2. Pantalla UI1 Inicio	48
4.3. Pantalla UI2 Espera de proceso	49
4.4. Pantalla UI3 Proceso concluido	51
4.5. Pantalla UI4 Resultados de consulta	53
4.6. Pantalla UI5 Cambio de periodo	53
4.7. Diagrama de secuencia	54
5.1. Etapas de Desarrollo	55
5.2. Etapas de la recolección	56
5.3. Ranking de sitios de noticias del período de enero del 2018 a enero del 2019.	57
5.4. Proceso de recolección	59
5.5. Noticias recolectadas durante el primer corte.	61
5.6. Noticias recolectadas al finalizar el segundo corte.	62
5.7. Noticias recolectadas por sitio web al finalizar el segundo corte	62
5.8. Noticias balanceadas por sección	64
5.9. Proceso de entrenamiento	65
5.10. Etapas de preprocesamiento	66
5.11. Corpus de entrenamiento	71
5.12. Corpus de prueba	71
5.13. Etapas de entrenamiento	72
5.14. Corpus de entrenamiento	78
5.15. Etapas de la aplicación web	81

5.16. Pantalla de Inicio.	83
5.17. Pantalla de espera	83

Índice de cuadros

3.1. Ejemplo de tokenización	18
5.1. Secciones de los sitios web	57
5.2. Ejemplo de estructura de un archivo CSV	60
5.3. Noticias recolectadas por sitio web	63
5.4. Identificador de sitio web	69
5.5. Etiquetas de secciones	75
5.6. Librería de algoritmo	75
5.7. Naive bayes	78
5.8. Máquina de soporte vectorial	79
5.9. Regresión logística	80
5.10. Random Forest	80

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 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** (Excélsior), 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 (?) 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
- Afinar el clasificador de noticias realizado en el trabajo terminal 2017-A02 para utilizarlo en el contexto de esta propuesta (filtro de sección)

¹www.animalpolitico.com

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 inglés y japonés, 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 clasificó 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 *Tecnology*; 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*
 - *Naive bayes classifier*
 - *Gaussean naive bayes*
 - *Multinomial naive bayes*
 - *Multivariate naive bayes*
 - *Bernulli naive bayes*

El mejor resultado es en combinación de *Term frequency-inverse document frequency* y *Multinomial naive bayes* con la precisión y exhaustividad mas alta el cual está alrededor de 98.4 % con un tiempo de 0.702 segundos, seguido

⁴Sitio web de noticias: www.ccnnindonesia.com

de *Term frequency-inverse document frequency* y *Bernulli naive bayes*(BNB) con 98.2 % en precisión y exhaustividad con un tiempo de .701 segundos.

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.

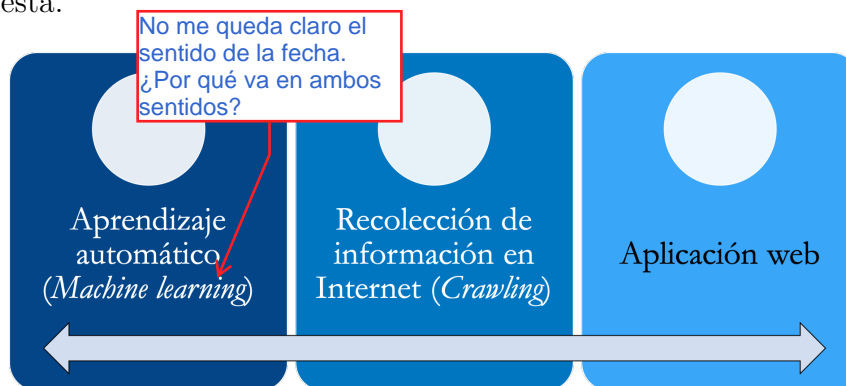


Figura 3.1: Campo de estudio

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 máquinas 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. Necesitamos una amplia variedad de métodos porque los datos basados en texto y en voz varían ampliamente, al igual que las aplicaciones prácticas.

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 **se generaliza** 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, mas a delante 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*), **Análítica 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 el proceso. El conjunto de palabras en el documento se convierten en *Sparse multidimensional representation*, 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 *y procesamiento del lenguaje natural*

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.

3.4.2. Pre-procesamiento

El Pre-procesamiento es necesario para convertir el formato no estructurado en un formato estructurado (Aggarwal, 2018). A menudo el texto contiene información extraña como etiquetas, *anchor text*¹, y otras características. En muchos casos las palabras son variaciones de otros (sinónimos) por el tipo de

¹Es el texto mostrado en los enlaces o hipervínculos, **Texto de anclaje** en español.

redacción, el contexto, para eliminar redundancia. Algunas palabras simplemente tienen faltas de ortografía. El proceso de convertir una secuencia de caracteres en una secuencia de palabras (tokens), es llamado “Tokenización”.

3.4.3. Extracción de texto

Cuando se recupera información de la web, se tiene que limpiar el texto ya que contiene etiquetas definidas por el hipertexto. Se debe 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).

3.4.4. Corpus

Se le llama corpus a la recopilación de un conjunto de textos, de materiales escritos y/o hablados, agrupados bajo un conjunto de criterios mínimos, para realizar ciertos análisis lingüísticos.

3.4.5. 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 ítems que se utiliza 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 hexadecimal, con el objeto de facilitar la identificación uno a uno cada carácter que compone la palabra. 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). Ejemplo (3.1): Hoy es un gran día para salir.

ID	1	2	3	4	5	6	7
Token	Hoy	es	un	gran	día	para	salir

Tabla 3.1: Ejemplo de tokenización

3.4.6. Lematización

Es el proceso lingüístico que, dada una palabra flexionada se encuentra su lema. Una palabra flexionada es cuando esta en el 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 anterior mente explicada
- Sintáctica: Toma encuentra el contexto donde se encuentra la palabra

Una opción para lematizar es *Freeling* (Padró and Stanilovsky, 2012), este es un lematizador hecho por la universidad de catalunia.

3.4.7. Frecuencia inversa del documento

Las palabras con poca frecuencia son mas discriminatorias que las de alta frecuencia. Por lo tanto se pondera la importancia de los documentos con base al calculo de la **frecuencia inversa del documento** (fid) en la colección. Si ψ_i es el número de documentos en el cual la palabra aparece, y ψ es el número total de documentos, la fid se calcula como $\log(\frac{\psi}{\psi_i})$. La importancia de un documento se calcula multiplicando la **frecuencia de término** (ft) en el documento por la fid . Mientras la ft brinda la cantidad de veces que una palabra aparece en el documento la fid especifica la importancia en la colección; Se define como **ft-fid** o *tf-idf* (Por su sigla en ingles *Term frequency – Inverse document frequency*) es la multiplicación de ft por idf (Aggarwal, 2018).

3.5. Aprendizaje supervisado

Los algoritmos de aprendizaje supervisado dependen de datos previamente etiquetado, es decir se necesita un corpus de datos, para llevar acabo el

entrenamiento, así el algoritmo pueda comprender los datos y con ello determinar que etiqueta debe asignarse a los nuevos datos en función del patron y asociando los patrones a los nuevos datos sin etiquetar. Después de ello, la maquina recibe un nuevo conjunto de datos para que el algoritmo de aprendizaje supervisado analice los datos y produzca el resultado de los datos etiquetados (CleverData, 2019).

Agregar al final "A continuación se describen algunos de los métodos más utilizados de Aprendizaje Automático aplicados a tareas de texto"

3.5.1. 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.

3.5.2. 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 *multinomial Naive Bayes*, 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 .

3.5.3. Maquina 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.

3.5.4. 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 muestras 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.

3.6. 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 $n \times n$,

		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

donde n es el número de clases diferentes con las que se están trabajando (Visa et al., 2011).

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:

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

Recolección de información de Internet

3.7. 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 datos estructurado². Los investigadores limpian y organizan el contenido para analizar la información.

3.7.1. 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.

²Un conjunto de datos estructurado permite recolectar varios valores simultáneamente.

- **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 las técnicas de Web scraping. La Figura 3.3 muestra los procesos.

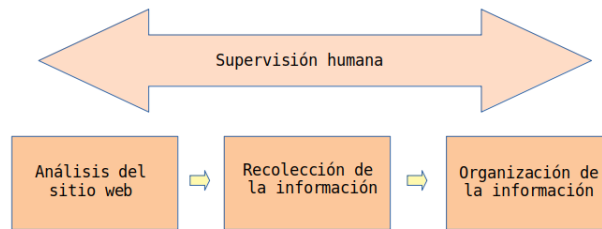


Figura 3.3: Etapas del proceso de *Web scraping*.

3.8. 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 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.

3.9. Python

*Python*³ 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

³<http://www.python.org/>

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](#)).

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

Aplicación web

3.10. 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 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 encuesta⁴ 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 infresar a distintos recursos.

3.10.1. 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,

⁴<https://www.inegi.org.mx/programas/dutih/2018/>

audio y video disponibles para cualquier persona con una computadora y una conexión a Internet.

3.11. 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 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](#)).

3.12. 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.

3.12.1. 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](#)).

3.12.2. 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](#)).

3.12.3. 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 ampliar 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.

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 clasificación



- **Descripción:** El tiempo de clasificación de las noticias recolectadas no debe tardar mas de cinco segundos.

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 15 noticias clasificadas, por los filtros seleccionados por el usuario.

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/>
- **Excelsior:** <https://www.excelsior.com.mx/>
- **La Jornada:** <https://www.jornada.com.mx/ultimas>
- **Milenio:** <https://www.milenio.com/>
- **Yahoo:** <https://es-us.noticias.yahoo.com/>
- **Sopitas:** <https://www.sopitas.com/>
- **SDP Noticias:** <https://www.sdpnoticias.com/>
- **Uno TV:** <https://www.unotv.com/inicio/>

- Referenciado por: CU1 Recolectar noticias

RN4 Umbral de grado de pertenencia



- **Descripción:** Solo se puede mostrar una noticia si su grado de pertenencia a una sección es mayor o igual al umbral establecido.
- Referenciado por: CU2 Clasificar noticias

RN5 Orden de publicación



- **Descripción:** Las noticias se muestran de forma descendente de acuerdo al grado de pertenencia a la sección.
- 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 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

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. Los casos de uso marcados en color gris son descritos en el documento, mientras que los mostrados en color rojo serán desarrollados en trabajo terminal II.

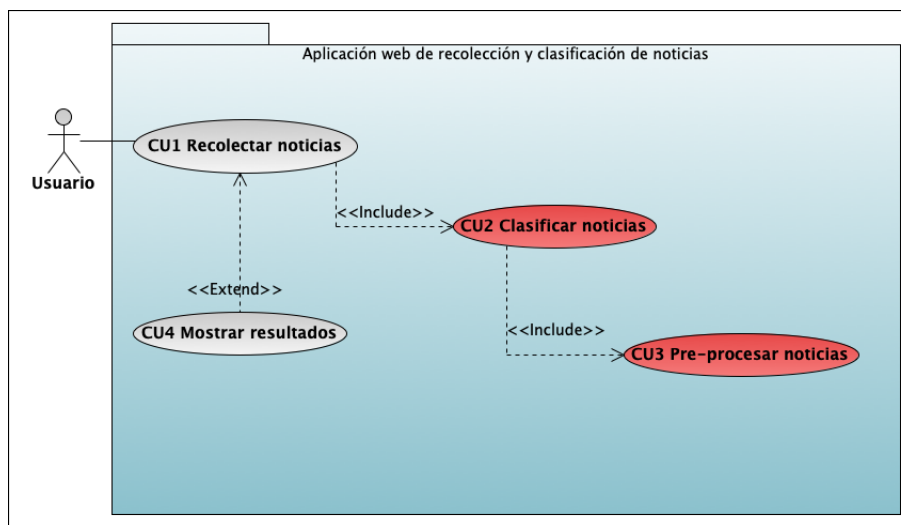


Figura 4.1: Diagrama de casos de uso

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. Se crea un proceso de recolección independiente por cada sitio web, para simular un ambiente de extracción en paralelo; 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**. Cabe destacar que las ligas contenidas en los sitios visitados son extraídas para su posterior análisis.

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:	<ul style="list-style-type: none"> • MSG1 Catálogo vacío • MSG3 Fallo en la recolección
Precondición:	EL Diccionario de URL'S debe contener los vínculos de los sitios a consultar
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
Errores:	<ul style="list-style-type: none"> • Uno: Cuando no se ha recuperado ninguna dirección web se muestra el mensaje MSG1 Catálogo vacío

Caso de uso:	CU1 Recolectar noticias
Errores:	<ul style="list-style-type: none"> • Dos: Cuando no se puede extraer información de los sitios brindados, se muestra el mensaje MSG3 <i>Fallo en la recolección</i>

Trayectoria principal

1.  Selecciona una opción de la pantalla **UI1 Inicio; Política, Economía, Deportes, Ciencia y tecnología o Cultura**.
2.  Obtiene las direcciones web con base en la regla de negocio **RN3 Listado de fuentes noticiosas**.
3.  Verifica que al menos se recupere una **Dirección web**. [Trayectoria A]
4.  Muestra la pantalla **Pantalla UI2 Espera de proceso**. [Trayectoria B]
5.  Verifica que no se haya recolectado noticias previamente. [Trayectoria C]
6.  Por cada URL recuperada se extraen las noticias con base en la regla de negocio **RN6 Periodo de recolección** y **RN7 Campos recolectados de noticia**. [Trayectoria D]
7.  Incluye el caso de uso **CU2 Clasificar noticias**.
8.  Ordena las noticias clasificadas de acuerdo a la regla de negocio **RN5 Orden de publicación**.
9.  Muestra la pantalla **Pantalla UI3 Proceso concluido**.
10. - - - - *Fin del caso de uso.*

Trayectoria alternativa A:



Condición: *No se ha recuperado ninguna dirección web* [Error Uno]

A-1.  Muestra el mensaje **MSG1 Catálogo vacío** en la pantalla **UI1 Inicio**.

A-2. - - - - *Fin del caso de uso.*


Trayectoria alternativa B:

Condición: *El usuario ha presionado el botón cancelar*

- B-1.  Presiona el botón **Cancelar** de la pantalla [Pantalla UI2 Espera de proceso](#).
- B-2.  Muestra la pantalla [UI1 Inicio](#).
- B-3. - - - - *Fin del caso de uso.*

Trayectoria alternativa C:

Condición: *Ya se han recolectado noticias*

- C-1.  Continúa en el paso [9](#) de la trayectoria principal.
- C-2. - - - - *Fin de la trayectoria.*

Trayectoria alternativa D:

Condición: *No se puede extraer información de los sitios web [\[Error Dos\]](#)*

- D-1.  Muestra el mensaje [MSG3 Fallo en la recolección](#) en la pantalla [UI1 Inicio](#).
- D-2. - - - - *Fin del caso de uso.*

Puntos de extensión

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

Región de la trayectoria: Proviene del paso [9](#) de la trayectoria principal.

Extiende a : [CU4 Mostrar resultados](#)

4.5.3. 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**. La consulta se realiza en un periodo establecido; el sitio muestra hasta 15 noticias, cada artículo contiene el **Título**, 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 información**.

Descripción

Caso de uso:	CU4 Mostrar resultados
Actor:	Usuario
Propósito:	Brindar una herramienta que permita consultar las noticias clasificadas
Entradas:	Periodo de búsqueda: Se selecciona con el mouse
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 ◦ Resumen
Precondición:	La clasificación de las noticias debe estar completa
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 **Aceptar** de la pantalla [Pantalla UI3 Proceso concluido](#). [[Trayectoria A](#)]

2. ● Obtiene la fecha actual.
3. ● Muestra hasta 15 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 información. [\[Trayectoria B\]](#)
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 hasta 15 noticias de las ordenadas previamente, 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 en el 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.*



4.5.4. CU3 Pre-procesar noticias

Resumen

Descripción

Caso de uso:	CU1 Recolectar noticias
Actor:	Usuario
Propósito:	
Entradas:	
Salidas:	
Precondición:	
Postcondiciones:	
Reglas de negocio:	
Errores:	

Trayectoria principal

1. 
2. 
3. - - - - *Fin del caso de uso.*



Trayectoria alternativa A:

Condición:

- A-1. 
- A-2. - - - - *Fin del caso de uso.*

Trayectoria alternativa B:

Condición:

- B-1. 
- B-2. 

B-3. - - - - *Fin del caso de uso.*

Puntos de extensión

Causa de la extensión:

Región de la trayectoria:

Extiende a :

4.6. Mensajes

MSG1 Catálogo vacío



- **Tipo:** Error.
- **Objetivo:** Dar a conocer que no se tiene las ligas a los sitios web.
- **Redacción:** El catálogo **Direcciones web** se encuentra vacío.
- **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

MSG3 Fallo en la recolección



- **Tipo:** Error.
- **Objetivo:** Informar al usuario que las ligas registradas en el diccionario de **URLs** no permiten extraer información.
- **Redacción:** No se puede extraer noticias de los sitios registrados en este portal web.
- **Referenciado por:** CU1 Recolectar noticias

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

- [MSG1 Catálogo vacío](#)
- [MSG3 Fallo en la recolección](#)

Comandos

1. **Inicio:** Te dirección 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 acabo. Además brinda un punto acceso para cancelar el proceso.

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. En la parte inferior se muestra el botón **Cancelar** el cual permite detener la consulta.

Salidas

- Ninguno

Comandos



Figura 4.3: Pantalla UI2 Espera de proceso

1. **Cancelar:** Detiene el proceso de recolección y clasificación, re-direcciona a la pantalla [UI1 Inicio](#).

Referenciado por

- [CU1 Recolectar noticias](#)

4.7.3. UI3 Proceso concluido

Objetivo

Informal 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 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**

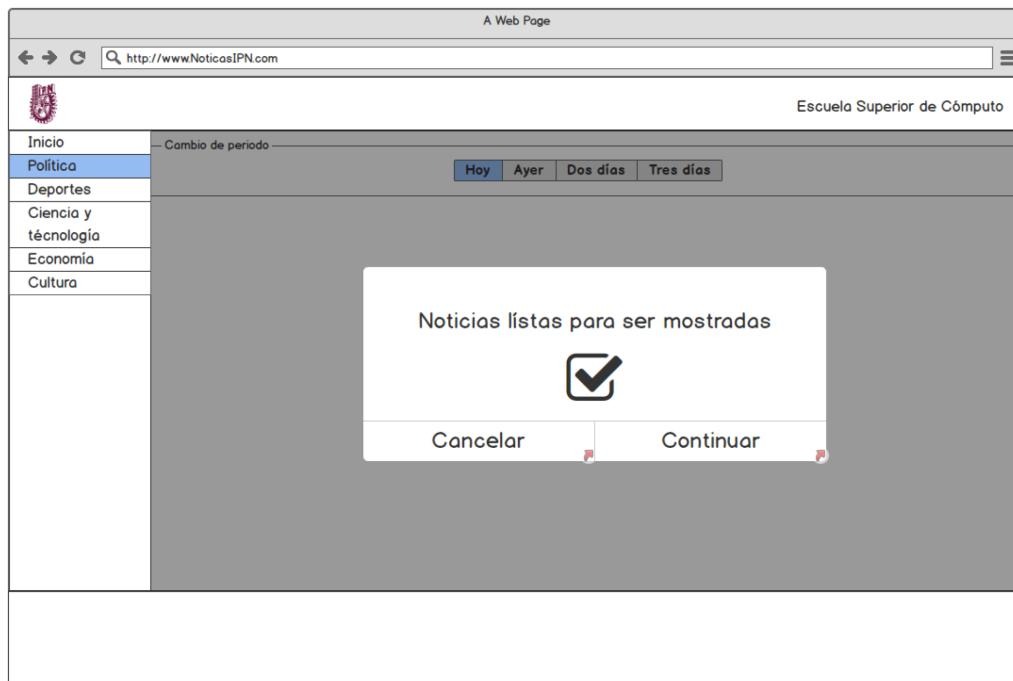


Figura 4.4: Pantalla UI3 Proceso concluido

- **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

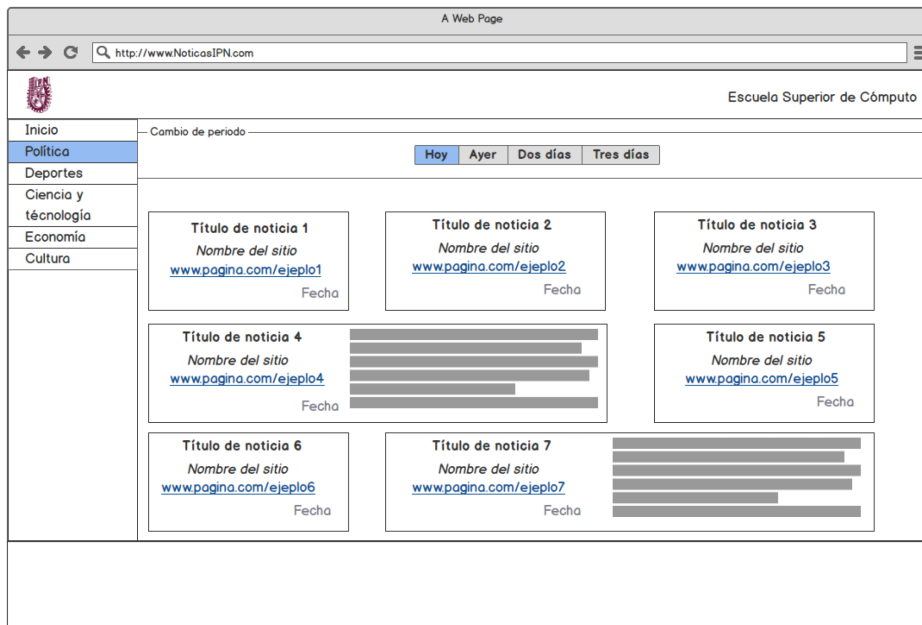


Figura 4.5: Pantalla UI4 Resultados de consulta

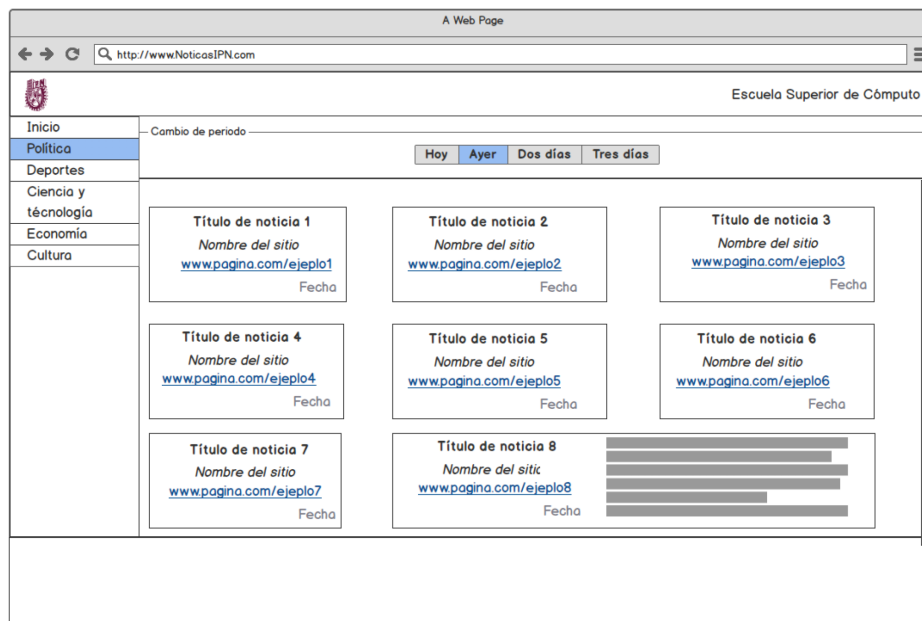


Figura 4.6: Pantalla UI5 Cambio de periodo

4.8. Diagrama de secuencia

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

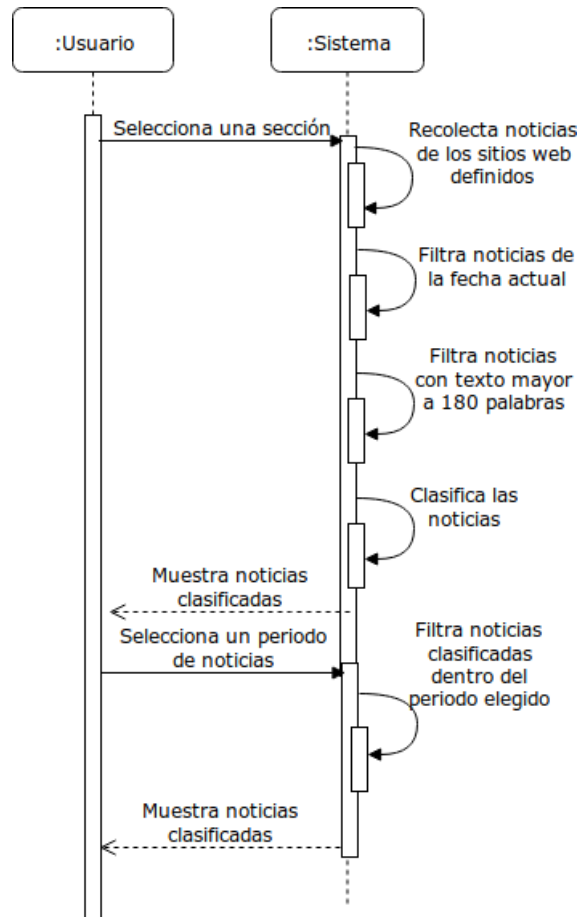


Figura 4.7: Diagrama de secuencia

Capítulo 5

Implementacion y Pruebas



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

recolección de noticias,

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

El sitio web El Economista¹ contiene una sección llamada **Ranking de Medios Nativos Digitales**², 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 5.3

¹<https://www.eleconomista.com.mx/>

²<https://www.eleconomista.com.mx/Ranking-de-Medios-Nativos-Digitales>

CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS 5.1. RECOLECCIÓN

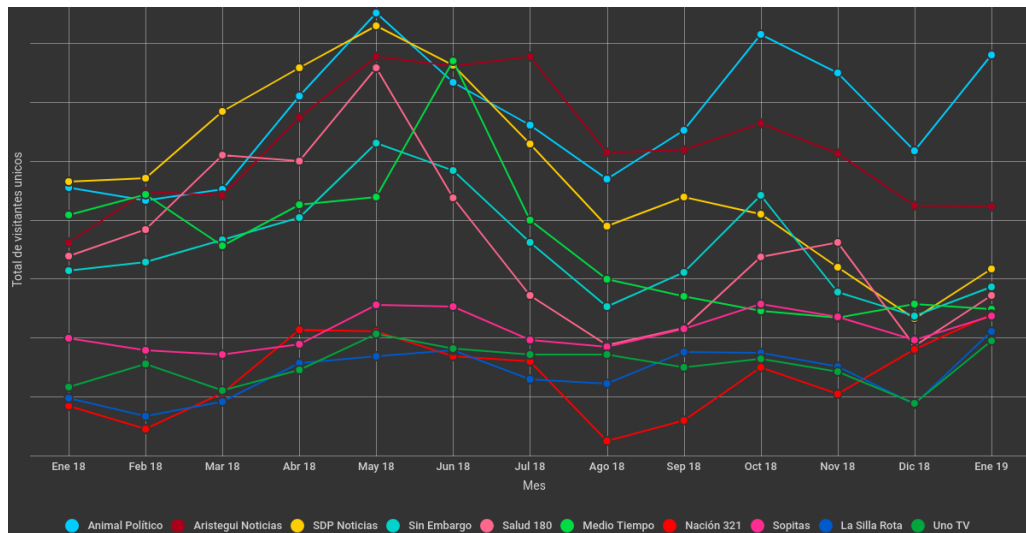


Figura 5.3: Ranking de sitios de noticias del período de enero del 2018 a enero del 2019.

Con base en la Figura 5.3 se han seleccionado los diarios Aristegui Noticias, El Economista, La Jornada, La Prensa, Proceso, Sopitas y TV Azteca para recolectar noticias. Los sitios organizan 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.

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

Tabla 5.1: Secciones existentes en los sitios web

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 cambiara.

Una expresión *XPath* de ruta permite buscar y seleccionar los distintos nodos de un documento XML(ver). En el siguiente Cuadro 5.1.1 se muestra un ejemplo con los elementos de una nota, los cuales son: **para**, **de**, **titulo**, **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>
  <titulo>Recordatorio</titulo>
  <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???) un *Crawler* te permite descargar información de una página web, como se muestra en la Figura 5.4. La im-

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

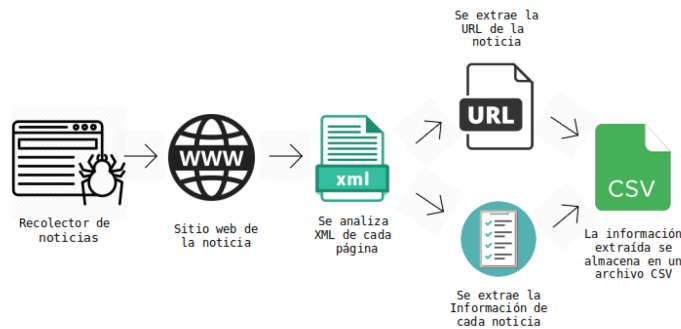


Figura 5.4: Proceso de recolección

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**³, en conjunto con **Scrapy**⁴, el cual es un framework que permite la extracción de información de sitios web.

El siguiente Cuadro 5.1.3, muestra los comandos utilizados para instalar las librerías utilizadas.

Cuadro 5.1.3: Comandos para instalar librerías utilizadas
<pre>\$ sudo apt-get install python3.6 \$ pip install Scrapy</pre>

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

³<https://www.python.org/>

⁴<https://scrapy.org/>

5.1. RECOLECCIÓN CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS

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

Cada uno de los recolectores contenía expresiones *XPath* que permitían recolectar la información de cada noticia, el Cuadro 5.1.4 muestra un ejemplo de las expresiones *XPath* utilizadas para recolectar noticias del sitio web Arístegui Noticias⁵

Cuadro 5.1.4: Ejemplo de expresiones *XPath* del sitio Arístegui 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 (ver **ver??**) 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<https://aristeguinoticias.com/>

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** 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 dividían las noticias recolectadas por sitio web.

Una vez finalizada la primera etapa de recolección, los resultados obtenidos por sección se muestran en la Figura 5.5.

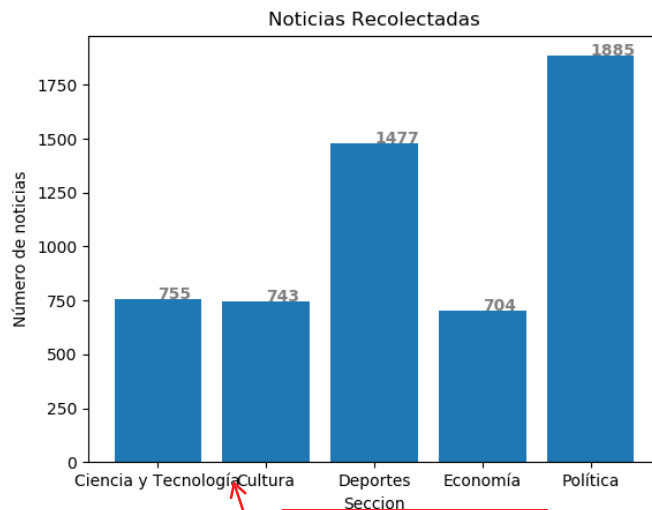


Figura 5.5: Noticias recolectadas durante el primer corte.

Cabe destacar que el número de noticias recolectadas durante el primer corte, no se encontraba **balanceado**, 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 5.6

5.1. RECOLECCIÓN CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS

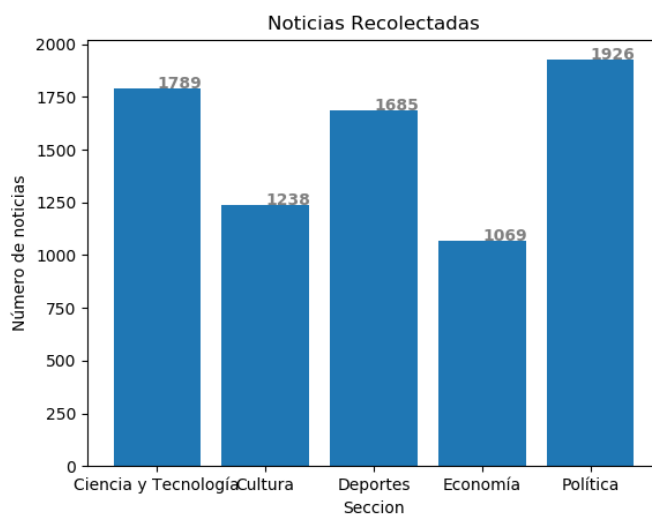


Figura 5.6: Noticias recolectadas al finalizar el segundo corte.

Los resultados que obtuvimos por sitio web se muestran en la Figura 5.7

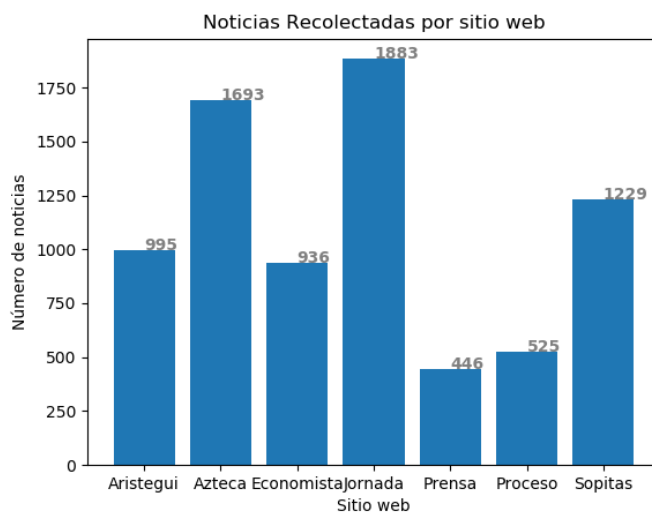


Figura 5.7: Noticias recolectadas por sitio web al finalizar el segundo corte

Los noticias recuperadas por cada sitio web, se muestran en la siguiente Tabla 5.1.

CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS 5.1. RECOLECCIÓN

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

5.1. RECOLECCIÓN CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS

Una vez concluida la recolección de noticias se procedió con eliminar aquellas noticias que hayan sido duplicadas, finalmente el número total de noticias por sección se muestra en la Figura 5.8, obteniendo un total de 3,500 noticias.

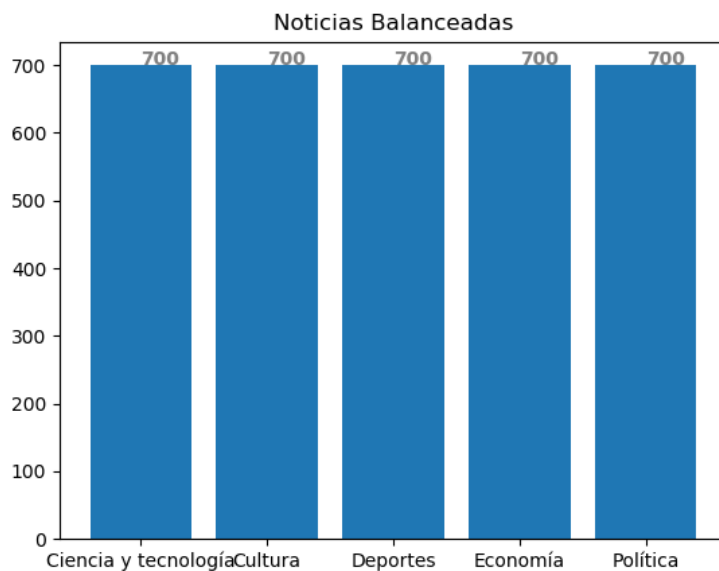


Figura 5.8: Noticias balanceadas por sección

Aquí falta la parte
donde explican como
normalizaron o
homologaron las
secciones de los
diferentes diarios

5.2. Entrenamiento de clasificador

El segundo pilar del trabajo terminal es entrenar un algoritmo (con aprendizaje supervisado), para clasificar las noticias en las secciones definidas en el capítulo 4 (ver). Cabe destacar que el clasificador resolverá un problema multiclase (ver) debido a que la entrada es un artículo y como salida brinda la pertenencia a una sección de 5 posibilidades. El proceso de entrenamiento se muestra en la Figura 5.9.

Como se mencionó en la etapa de recolección, el corpus ocupado contiene 700 noticias por cada sección es decir, 3500 artículos como total del dataset. Sin embargo solo el 90 % de este será ocupado en la etapa de entrenamiento y el 10 % restante se ocupará para hacer la etapa de prueba, teniendo así un total de 3,150 noticias de entrenamiento y 350 para medir la eficiencia del clasificador, estos conjuntos serán definidos al final de la etapa de preprocesamiento.

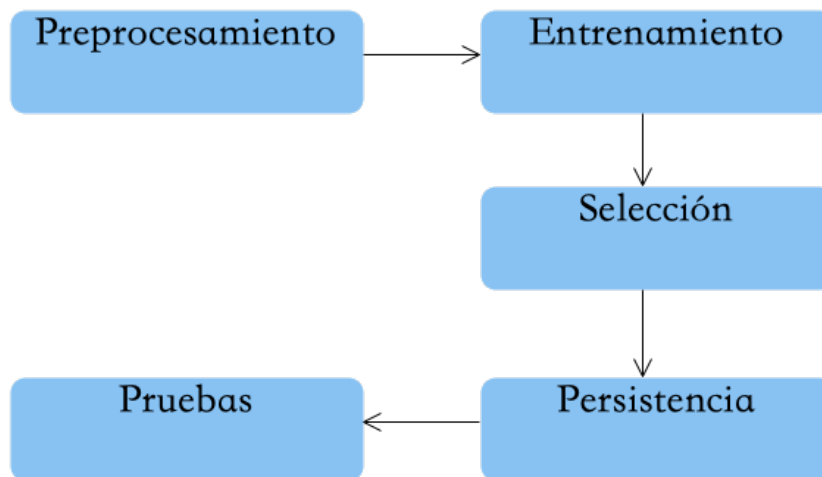


Figura 5.9: Proceso de entrenamiento

5.2.1. Preprocesamiento

Como primera instancia, el corpus creado de 3500 noticias debe ser procesado, con el fin de crear vectores que representan el contenido de cada artículo

de forma ordenada **(ver)**, 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 5.10.

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

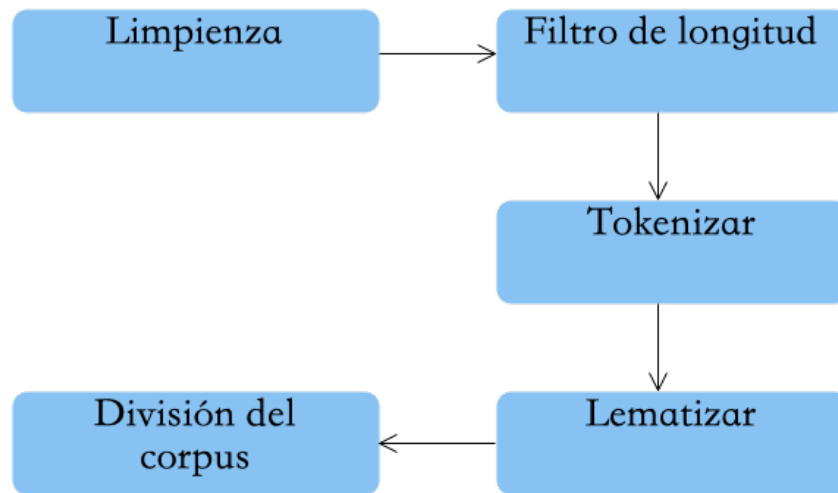


Figura 5.10: Etapas de preprocesamiento

Limpieza

Esta etapa consiste en eliminar texto que no brinda información útil para el entrenamiento como, hipertexto (ver), 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 ocupando las librerías; **pandas** **(ver)** como medio de lectura de archivos tipo **CSV** (*comma separeted values*, por sus

CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS. ENTRENAMIENTO

siglas en ingles); **re** (ver) la cual permite evaluar expresiones regulares con el objetivo de eliminar símbolos; **demoji** (ver) 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.

Filtro de longitud

Con base en la regla de negocio **(ver)** 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.

Tokenizar

La etapa de tokenización consiste en separar el texto en sus elementos míni-

mos 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 **ocupo** la librería **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 .

Lematizar

Lematizar es el proceso de reducir cada palabra a su **lemma** con el fin de **eliminar ruido** en el texto, por ejemplo las palabras correrás, corriendo, corré, tienen como lema el verbo correr, el plural niños tiene como lema niño (**ver**). Para realizar esta tarea se ha usado **spacy** (**ver**) 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

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

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.

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

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.

Cuadro 5.2.5: Estructura de archivo

```
id&&&&título&&&&noticia&&&&seccion
1001&&&&Titulo 1&&&&Contenido noticia 1&&&&0
...
5003500&&&&Titulo 3500&&&&Contenido noticia 3500&&&&4
```

División del corpus

Para el correcto diseño y evaluación del algoritmos se requiere dividir el corpus en dos conjuntos: **entrenamiento** y **prueba**, con un 90 % y 10 %

del total del **dataset** 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 **ocupado** una técnica llamada *Shuffle*.

Shuffle consiste en brindar un arreglo con los identificadores de las noticias y un número (nombrado usualmente como semilla), quien genera un nuevo orden en los identificadores de acuerdo a los números pseudo aleatorios que retorna esta función, tomando este como el nuevo orden de los textos en el archivo de almacenamiento. Para el desarrollo de esta etapa se ha utilizado la librería **Shuffle** (ver) 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 3150) y de prueba (con 350).

La cantidad de noticias por sección del conjunto de entrenamiento se muestran en la Figura 5.11 y del conjunto de prueba en la Figura **??**.

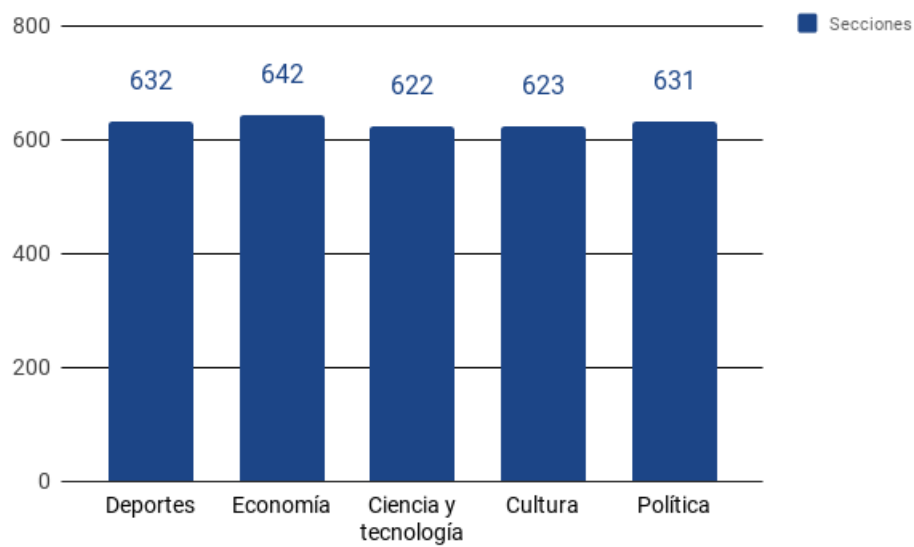


Figura 5.11: Corpus de entrenamiento

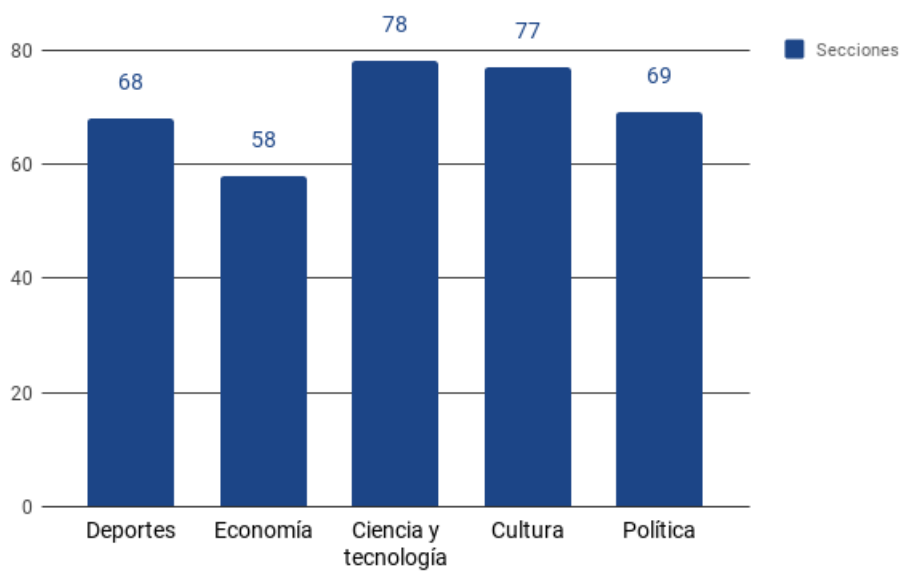


Figura 5.12: Corpus de prueba

5.2.2. Entrenamiento

Para crear un modelo clasificador (el cual resuelve un problema multinomial ver) usando aprendizaje supervisado (ver), 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 dataset serán usados en 4 algoritmos (seleccionados con base en el estado del arte ver), los cuales son:

- Naive Bayes (ver)
- Regresión logística (ver)
- Maquina de soporte vectorial (ver)
- Random Forest (ver)

El proceso de entrenamiento consta de 5 pasos los cuales se muestran en la Figura 5.13.

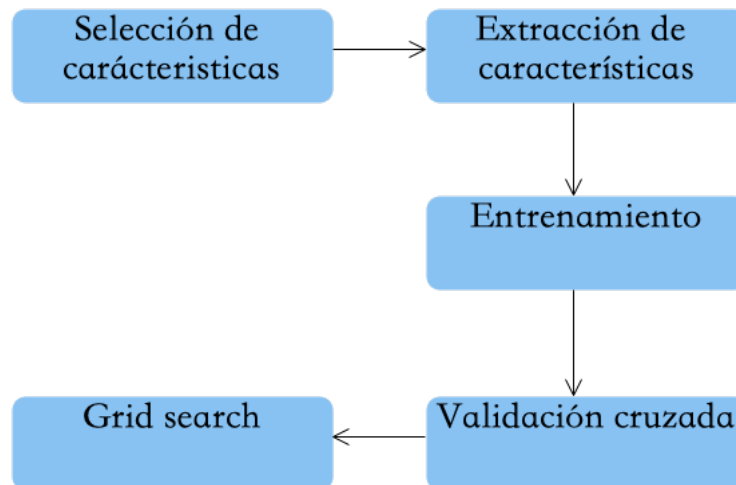


Figura 5.13: Etapas de entrenamiento

El desarrollo se ha implementado en el lenguaje de programación **Python 3**, ocupando la librería **scikit learn** quien permite crear instancias de los algoritmos mencionados. Cabe señalar que el objetivo de este trabajo terminal es el entrenamiento de los clasificadores y no su desarrollo.

Como primer paso del entrenamiento, el corpus debe ser traducido del lenguaje español a un conjunto de vectores numéricos que representan la información de las noticias, esta es llamada representación vectorial (ver). Para lograr este objetivo, del *dataset* se deben seleccionar características que definan los elementos del vector numérico.

Selección 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 selección de características es el proceso de tomar el corpus de noticias e identificar las palabras mas comunes en los artículos.

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 las palabras relevantes las cuales son mostradas 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>Esta</i>	<i>noticia</i>	<i>es</i>	<i>la</i>	<i>segunda</i>	<i>noticia</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

<i>es</i>	<i>esta</i>	<i>este</i>	<i>la</i>	<i>noticia</i>	<i>primera</i>	<i>segunda</i>	<i>tercera</i>
-----------	-------------	-------------	-----------	----------------	----------------	----------------	----------------

Extracción de características

Después de seleccionar las características, 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 implementado la **librería CountVectorizer** quien permite generar la selección y extracción de características. Cabe destacar que esta es la presentación vectorial (de cada noticia) que los algoritmos de clasificación pueden entender y por ende ser entrenados.

Cuadro 5.2.10: Extracción por frecuencia

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

Cuadro 5.2.11: Extracción binaria

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

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

CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS. ENTRENAMIENTO

semántica, esta técnica es llamada bolsa de palabras (ver).

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**, **Maquina de soporte vectorial**, **Random Forest**. Cada clasificador **reciben** como entrada un conjunto de vectores etiquetados y como salida se genera un modelo el cual predice la sección de nuevos textos.

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 ocupado una instancia de cada algoritmo, para esto se incluye la **librería** correspondiente de **scikitlearn**, las cuales se muestran en la Tabla 5.6, la entrada al clasificador es el conjunto de espacio vectorial y las etiquetas correspondientes, esto regresa como resultado un modelo que es **capas** de predecir la sección de noticias, sin que estas estén etiquetadas.

Sección	Etiqueta
Naive Bayes	MultinomialNB
Maquina de soporte vectorial	SVC
Regresión logística	LogisticRegression
Ramdon Forest	RandomForestClassifier

Tabla 5.6: Librería de algoritmo

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.

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 **precisión** 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 **precisión**. En términos generales este promedio nos brinda mayor confianza en el resultado del entrenamiento de cada clasificador (para ver una explicación mas detallada ver).

Variación 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. A continuación se explican **las entradas** de cada clasificador:

Naive Bayes

El parámetro en el cual se varía en este algoritmo, es un escalar llamado **Alpha** (α). Como se explico en el capítulo 3 (ver), un valor numérico es asignado a cada palabra en el corpus con respecto a la frecuencia de aparición en un clase (como se muestra en la ecuación 5.1). El valor α evita que dicha ecuación se haga cero, en el caso de la ausencia de una palabra en una clase c (es decir $N_{ck} = 0$).

$$P(t_k|c) = \frac{N_{ck} + \alpha}{N_c + \alpha n} \quad (5.1)$$

Regresión logística

Este algoritmo está basado en una regresión lineal y el calculo de probabilidades como se explica en el capítulo 3(ver). Existen varias **formas de implementar** este algoritmo, en este trabajo se ha implementado de dos formas: optimizando la función l_1 (ver 5.2) y minimizando el costo de la función l_2 (ver 5.3). Como se observa ambas ecuaciones contiene el escalar C , 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.

$$\min_{w,c} |w|_1 + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1) \quad (5.2)$$

$$\min_{w,c} |w|_1 + C \sum_{i=1}^n \log(\exp(-y_i(X_i^T w + c)) + 1) \quad (5.3)$$

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 : $[1e - 6, 1e - 05, 1e - 04, 1e - 03, 1e - 02, 1e - 01]$.

← Explicar los parámetros de los demás algoritmos

Ahora que se han explicado los parámetros de cada algoritmo, podemos calcular el mejor. Esta etapa consiste en variar los parámetros y aplicar validación cruzada, con el objetivo de conseguir el mejor clasificador y los mejores parámetros. Para ejemplificar esta tarea la Figura 5.14 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.

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.

Las pruebas de este trabajo terminal han implementado 5 pliegues en la validación cruzada y se ha usado el corpus con 3150 noticias como entrenamiento.

La Tabla 5.7 muestra los resultados de la variación del parámetro **Alpha** en 0.5, 1.0, 1.5 y 2.

```
[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 5.14: Corpus de entrenamiento

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

CAPÍTULO 5. IMPLEMENTACION Y PRUEBAS. ENTRENAMIENTO

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 5.8: Maquina de soporte vectorial

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 5.9: 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 5.10: Random Forest

5.2.3. Selección

5.2.4. Pruebas

5.2.5. Persistencia

5.3. Aplicación web

La siguiente Figura 5.15 muestra las etapas que se llevarán a cabo por parte de la aplicación web.

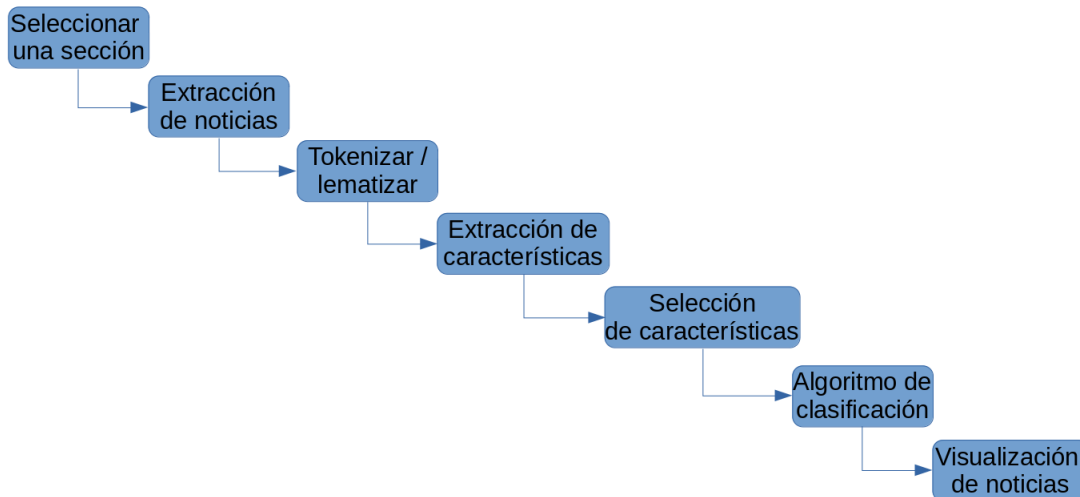


Figura 5.15: Etapas de la aplicación web

5.3.1. Tecnologías utilizadas

A continuación se describen las tecnologías y librerías que han sido utilizadas durante el desarrollo del sistema Web.

JavaServer Faces

JavaServer Faces es una tecnología utilizada para desarrolladar aplicaciones web la cual permite el desarrollo de interfaces de usuario.

5.3.2. Proceso de recolección

La aplicación web tiene como objetivo recolectar y clasificar noticias, por ello la primera parte que se debe realizar es la recolección de noticias, como

se mencionó en el apartado 5.1 se han definido los sitios web de donde las noticias serán recolectadas, la información recuperada **será:**

- URL de la noticia
- Título
- Fecha
- Autor
- **Descripción**⁶
- Noticia

Cabe destacar que en este punto, ya no será necesario extraer el nombre de la sección a la cual pertenece la noticia, dado que el clasificador lo realizará de manera automática.

5.3.3. Proceso de clasificación

Una vez que las noticias han sido recolectadas, deberán pasar por dos pasos fundamentales, como primer paso la noticia debe ser **tokenizada y lematizada**, posteriormente se deben extraer las características de cada noticia, esto permitirá al algoritmo seleccionado obtener un mejor resultado en el momento de realizar la clasificación.

5.3.4. Frontend

Para la realización de la aplicación web es necesario utilizar herramientas que nos permitan visualizar las noticias clasificadas, por ello las interfaces de usuario se desarrollarán en utilizando el Framework Java Server Faces.

La Figura 5.16 muestra la vista que tendrá el usuario al ingresar a la aplicación web

El usuario tiene la posibilidad de elegir la sección de la cual desee visualizar noticias. La Figura 5.17 muestra la vista que tendrá el usuario dar click en una sección.

Una vez finalizado el proceso de recolección de noticias

⁶Existen sitios web, donde no todas las noticias cuentan con una descripción



Figura 5.16: Pantalla de Inicio.



Figura 5.17: Pantalla de espera

Bibliografía

- Aggarwal, C. C. (2018). *Machine Learning For Text*. Springer, primera edición.
- 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.
- 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/>.
- 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 edition.
- 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."
- Padró, L. and Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality. pages 2473–2479. European Language Resources Association (ELRA).
- Pajares, G. and Santos, M. (2006). *Inteligencia artificial e ingeniería del conocimiento*. Alfaomega.

- 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.