



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

**CARRERA DE INGENIERÍA EN SISTEMAS E
INFORMÁTICA**

**TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SISTEMAS E INFORMATICA**

**“SISTEMA DE EXTRACCIÓN DE CARACTERÍSTICAS Y
ANÁLISIS DE TEXTO LÍRICO MUSICAL EN CONTENIDOS WEB
MEDIANTE APRENDIZAJE DE MÁQUINAS PARA BRINDAR
RECOMENDACIONES A COMPOSITORES LATINOAMERICANOS”**

AUTOR: REIMUNDO GUALOTUÑA, EVELYN ANDREA

DIRECTOR: ING. GUERRERO IDROVO, ROSA GRACIELA MSC.

SANGOLQUI

2018

CERTIFICADO



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, “SISTEMA DE EXTRACCIÓN DE CARACTERÍSTICAS Y ANÁLISIS DE TEXTO LÍRICO MUSICAL EN CONTENIDOS WEB, MEDIANTE APRENDIZAJE DE MÁQUINAS PARA BRINDAR RECOMENDACIONES A COMPOSITORES LATINOAMERICANOS” fue realizado por la señorita Reimundo Gualotuña Evelyn Andrea, el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 26 de febrero de 2018.

Firma: 
Ing. Graciela Guerrero MSc.
C. C. 1720513322

AUTORÍA DE RESPONSABILIDAD**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN****CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA****AUTORÍA DE RESPONSABILIDAD**

Yo, Evelyn Andrea Reimundo Gualotuña, con cédula de identidad N°1720181286, declaro que este trabajo de titulación "SISTEMA DE EXTRACCIÓN DE CARACTERÍSTICAS Y ANÁLISIS DE TEXTO LÍRICO MUSICAL EN CONTENIDOS WEB MEDIANTE APRENDIZAJE DE MÁQUINAS PARA BRINDAR RECOMENDACIONES A COMPOSITORES LATINOAMERICANOS" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas. Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 26 de febrero del 2018

Evelyn Andrea Reimundo Gualotuña

C.C: 1720181286

AUTORIZACIÓN



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERIA EN SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

Yo, Reimundo Gualotuña, Evelyn Andrea autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"SISTEMA DE EXTRACCIÓN DE CARACTERÍSTICAS Y ANÁLISIS DE TEXTO LÍRICO MUSICAL EN CONTENIDOS WEB MEDIANTE APRENDIZAJE DE MÁQUINAS PARA BRINDAR RECOMENDACIONES A COMPOSITORES LATINOAMERICANOS"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 26 de febrero del 2018



Evelyn Andrea Reimundo Gualotuña
C.C: 1720181286

DEDICATORIA

El presente trabajo dedico en primer lugar a Dios por darme salud a través de estos años, otorgándome las fuerzas necesarias para saber sobrellevar las dificultades en el transcurso de mi vida.

A mis padres Wilson y Paulina, por el sacrificio que realizan día a día en sus actividades, por darme la oportunidad de llevar a cabo mis estudios universitarios, a mis hermanas quienes supieron guiarme y apoyarme en las buenas y las malas, por los consejos que me ayudaron a no desvanecer y seguir adelante en mi desarrollo profesional, pero en especial a mi hija Ariana Carolina, quien es mi principal fuente de inspiración para lograr mis metas y objetivos planteados a lo largo de mi vida, este logro es por ti y para ti pequeñita mía.

AGRADECIMIENTO

Agradezco a Dios por otorgarme las fuerzas necesarias para culminar con mi desarrollo profesional demostrando mis conocimientos y habilidades implementados en el presente trabajo de titulación, pero, sobre todo, gracias por darme al pilar fundamental que sostiene mi vida, y esa es mi hija Ariana Carolina.

A mis profesores de tan prestigiosa universidad que me impartieron sus conocimientos a lo largo de mis estudios que hoy llegan a su fin, en especial a mi directora Ing. Graciela Guerrero que confió en mis habilidades y supo guiarme dentro del proceso de desarrollo de mi investigación.

Un agradecimiento eterno a mis padres por confiar en mí, por darme todas las facilidades para finalizar mis estudios universitarios, a mis hermanas Diana y Brenda por las palabras de aliento que me supieron brindar para no desmayar y lograr mis metas.

Gracias a todas las personas que de una u otra manera forman parte de mi vida y han sabido colaborarme para poder culminar mis actividades académicas.

ÍNDICE

CERTIFICADO	i
AUTORIZACIÓN	iii
DEDICATORIA.....	iv
AGRADECIMIENTO	v
ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS	x
RESUMEN.....	xiii
ABSTRACT.....	xiv
CAPÍTULO 1	1
INTRODUCCION.....	1
1.1. Antecedentes.....	1
1.2. Planteamiento del problema.....	3
1.3. Justificación.....	4
1.4. Objetivos.....	6
1.5. Alcance.....	6
1.6. Hipótesis.....	8
1.7. Estructura del contenido	9
CAPÍTULO 2	11
MARCO TEÓRICO Y TRABAJOS RELACIONADOS.....	11
2.1. Marco Teórico.....	11
2.2. Extracción de características.....	12
2.2.1. Web Crawler	13
2.3. Análisis de texto lírico musical.....	14
2.3.1. Evaluación de hipótesis	14

2.3.2. Comparación de letras por género.....	15
2.3.3. Tendencias reveladoras.....	17
2.3.4. Captura de datos sociales.....	19
2.3.5. Recolección de comentarios.....	19
2.4. Inteligencia Artificial.....	20
2.4.1. Técnicas de Inteligencia Artificial.....	21
2.5. Aprendizaje automático.....	24
2.5.1. Métodos de Aprendizaje.....	25
2.5.2. Algoritmos y métodos de aprendizaje.....	28
2.5.3. Aplicaciones del aprendizaje automático.....	28
2.6. SCRUM.....	30
2.6.1. Proceso Scrum.....	31
2.6.2. Roles.....	31
2.6.3. Aplicación.....	32
2.7. WEKA.....	33
2.8. Trabajos Relacionados.....	35
2.9. Conclusiones.....	39
CAPITULO 3.....	41
DISEÑO.....	41
3.1. Arquitectura.....	41
3.2. SCRUM.....	45
3.2.1. Planificación.....	45
3.2.2. Sprint 1: Web Crawler.....	48
3.2.3. Sprint 2: Módulo Weka.....	55
3.2.4. Sprint 3: Módulo Página Web.....	59
3.3. Maquetado del sistema.....	63

3.4. Desarrollo de software.....	65
3.5. Conclusiones.....	70
CAPITULO 4.....	72
PRUEBAS Y ANÁLISIS DE RESULTADOS.....	72
4.1. Pruebas de Análisis de Datos.....	73
4.1.1. Discusión de resultados.....	81
4.2. Pruebas funcionales del sistema.....	83
4.3. Tabulación de la Encuesta.....	86
CAPITULO V.....	89
CONCLUSIONES Y RECOMENDACIONES.....	89
5.1. Conclusiones.....	89
5.2. Recomendaciones.....	90
5.3. Líneas de trabajos futuros.....	90
BIBLIOGRAFIA.....	92

ÍNDICE DE TABLAS

Tabla 1	<i>Aplicaciones del aprendizaje automático</i>	29
Tabla 2	<i>Técnicas del proceso de desarrollo</i>	39
Tabla 3	Definición de prioridades	47
Tabla 4	<i>Backlog del producto</i>	48
Tabla 5	<i>Requerimiento funcional 1</i>	49
Tabla 6	<i>Requerimiento funcional 2</i>	49
Tabla 7	<i>Historia de usuario sprint 1</i>	50
Tabla 8	<i>Especificación de caso de uso: seleccionar url</i>	52
Tabla 9	<i>Especificación de caso de uso- extraer información</i>	52
Tabla 10	<i>Especificación de caso de uso-descargar archivo</i>	53
Tabla 11	<i>Backlog sprint 1-módulo web crawler</i>	54
Tabla 12	<i>Requerimiento funcional 3</i>	55
Tabla 13	<i>Historia de usuario sprint 2</i>	56
Tabla 14	<i>Especificación de caso de uso-leer y analizar archivos planos</i>	57
Tabla 15	<i>Backlog sprint2: módulo weka</i>	58
Tabla 16	<i>Requerimiento funcional 4</i>	59
Tabla 17	<i>Historia de usuario sprint 3</i>	60
Tabla 18	<i>Especificación de caso de uso-diseñar y elaborar página web</i>	61
Tabla 19	<i>Backlog sprint 3-módulo página web</i>	63
Tabla 20	<i>Caso de prueba módulo web crawler</i>	84
Tabla 21	<i>Caso de prueba módulo weka</i>	84
Tabla 22	<i>Caso de prueba módulo aplicación web</i>	85

ÍNDICE DE FIGURAS

Figura 1 Arquitectura del Sistema.....	7
Figura 2 Arquitectura de un Web Crawler.....	8
Figura 3 Proceso de análisis de texto lírico	8
Figura 4 Proceso de reconocimiento de género musical	15
Figura 5 Comando de extracción de texto	16
Figura 6 Funcionamiento de la enciclopedia MusicBrainz.	17
Figura 7 Tendencias por lista de amigo en redes sociales	18
Figura 8 Técnicas de la Inteligencia Artificial	22
Figura 9 Evolución de la inteligencia Artificial	25
Figura 10 Modelo No Supervisado- Agrupamiento	26
Figura 11 Modelos Supervisado: Clasificación	27
Figura 12 Ciclo de aplicación SCRUM.....	31
Figura 13 Interfaz de Herramienta Weka	34
Figura 14 Arquitectura de un buscador centralizado.	42
Figura 15 Arquitectura Web Crawler.....	43
Figura 16 Arquitectura Weka	44
Figura 17 Arquitectura Web	45
Figura 18 Planificación del proyecto	46
Figura 19 Diagrama de procesos.....	47
Figura 20 Casos de Uso-Módulo Web Crawler.....	51
Figura 21 Modelo de Datos.....	53
Figura 22 Casos de Uso-Módulo Weka	57
Figura 23 Modelos de Datos.....	58
Figura 24 Casos de Uso-Módulo Página Web.....	61

Figura 25 Diseño Página Web	62
Figura 26 Maquetación del sistema- Género Salsa	64
Figura 27 Maquetación del sistema-Género Baladas	64
Figura 28 Función Crawler.....	65
Figura 29 Búsqueda de url's	66
Figura 31 Función Indexar	66
Figura 32 Función Buscar.....	67
Figura 33 Función Indexar	67
Figura 34 Resultados Lucene	68
Figura 35 Frecuencia de datos	69
Figura 36 Análisis de texto-Preprocesamiento Weka	69
Figura 37 Clúster Algoritmo K-means.....	70
Figura 38 Simple K-means	73
Figura 39 Diagrama Algoritmo SimpleKmean.....	74
Figura 40 Algoritmo HierarchicalClusterer	75
Figura 41 Diagrama Algoritmo HierarchicalClusterer.....	75
Figura 42 Algoritmo Cobweb	76
Figura 43 Diagrama Algoritmo Cobweb	77
Figura 44 Algoritmo EM	77
Figura 45 Diagrama Algoritmo EM.....	78
Figura 46 Algoritmo FarthestFirst	79
Figura 47 Diagrama Algoritmo FarthestFirst.....	80
Figura 48 Algoritmo Canopy	80
Figura 49 Diagrama del Algoritmo Canopy	81
Figura 50 Algoritmos Método Clúster Weka	82
Figura 51 Funcionamiento del Algoritmo Canopy	82

Figura 52 Funcionamiento del Algoritmo Simple Kmeans	83
Figura 53 Valoración Dimensión Antecedentes	86
Figura 54 Valoración Dimensión Usabilidad	87
Figura 55 Valoración Dimensión Aceptación	88

RESUMEN

El procesamiento del lenguaje natural es un campo que combina las ciencias computacionales como el aprendizaje automático con la lingüística aplicada, con el fin de lograr el procesamiento asistido por computador de cierta información expresada en lenguaje corporal para llevar a cabo ciertas tareas que imitan comportamientos inteligentes. Por esta razón, y en base a los procesos de aprendizaje que tiene el ser humano, se puede llegar a implementar el reconocimiento de texto utilizando las computadoras y el potencial analítico de algoritmos que forman parte de la herramienta Weka, minimizando el error en la clasificación de texto. Para ello, el desarrollo del prototipo propuesto en el trabajo de titulación parte de un estudio de características que contiene la web, en base a los recursos seleccionados se llegó a la construcción de un web crawler que permite la extracción y almacenamiento de texto por género musical de artistas latinoamericanos. Seguidamente, se muestra un análisis de los algoritmos que forman parte de la clusterización de datos indicando los porcentajes de instancias correctamente agrupadas. Para finalizar con el proceso de desarrollo, se elabora una página web que contiene el resultado del estudio que se llevó a cabo dentro de la investigación, logrando brindar recomendaciones en base a las canciones más populares de cada género musical.

PALABRAS CLAVE

- **PROCESAMIENTO DE LENGUAJE NATURAL**
- **APRENDIZAJE AUTOMATICO**
- **WEKA**
- **CLUSTERIZACION**
- **CRAWLER**

ABSTRACT

Natural language processing is a field that combines computational science such as machine learning with applied linguistics, in order to achieve computer-assisted processing of certain information expressed in body language to carry out certain tasks that mimic intelligent behavior. For this reason and based on the learning processes that the human being has, text recognition can be implemented using computers and the analytical potential of algorithms that are part of the Weka tool, minimizing the error in the classification of text. For this, the development of the prototype proposed in the titration work is based on a study of features contained in the web. Based on the selected resources, a web crawler was built that allows the extraction and storage of text by musical genre. of Latin American artists. Next, an analysis of the algorithms that are part of the clustering of data indicating the percentages of correctly grouped instances is shown. To finish the development process, a web page is created that contains the results of the study that was carried out within the research, making recommendations based on the most popular songs of each musical genre.

KEYWORDS:

- **PROCESSING OF NATURAL LANGUAGE**
- **AUTOMATIC LEARNING**
- **WEKA**
- **CLUSTERIZATION**
- **CRAWLER**

CAPÍTULO 1

INTRODUCCION

1.1. Antecedentes.

Una de las tareas más desafiantes en la ciencia de la computación es construir máquinas o programas de computadoras que sean capaces de aprender (García Cambroner & Gómez Moreno, 2009). El darles la capacidad de aprendizaje a las máquinas abre una amplia gama de nuevas aplicaciones. El entender también como éstas pueden aprender nos puede ayudar a entender las capacidades y limitaciones humanas de aprendizaje (García Cambroner & Gómez Moreno, 2009). En general, se busca construir programas que mejoren automáticamente con la experiencia. El aprendizaje no solo se encarga de obtener el conocimiento, sino también la forma en que este se representa.

El principal objetivo del Aprendizaje de Máquina (ML por sus siglas en inglés) es el desarrollo de sistemas que puedan cambiar su comportamiento de manera autónoma basados en su experiencia. ML ofrece algunas de las técnicas más efectivas para el descubrimiento de conocimiento (patrones) en grandes volúmenes de datos. ML ha jugado un rol fundamental en áreas tales como la bioinformática, la recuperación de información en la web, la inteligencia de negocios y el desarrollo de vehículos autónomos, etc. (Berenzweig, Logan, P.W. Ellis, & Whitman, 2003).

Cuando los seres humanos nos enfrentamos a un nuevo concepto que queremos aprender, nuestro cerebro no lo hace de forma aislada, sino que utiliza todo el conocimiento previamente aprendido para ayudarse en este nuevo aprendizaje. Además, nuestro cerebro es capaz de aislar lo que no va a beneficiarnos y a utilizar lo que realmente nos va ser útil, esto lo hace muy bien y de forma inconsciente. Sin embargo, cuando una máquina de aprendizaje es entrenada para resolver una determinada tarea, por ejemplo, a diagnosticar una determinada enfermedad, normalmente esta máquina aprende sólo con los datos disponibles sobre esa enfermedad (Bueno Crespo, 2013).

En sus orígenes, Internet era un espacio al que pocos usuarios podrían acceder, estaba especialmente controlado por grandes empresas y en su mayoría eran los que generaban y publicaban contenidos. Con la evolución de la Web surgen cambios que se orientan a una mayor interacción por parte de los usuarios finales, que ya no sólo consumen contenidos, sino que también tienen la oportunidad de generarlos. A medida que los usuarios han ido proporcionando contenidos se ha ido produciendo una progresiva, sobrecarga de información. La llamada web semántica o web inteligente, se basa en la manipulación de datos más eficiente a través de datos semánticos. La web 3.0 tiene como protagonista al procesador de la información (máquina) que debe ser capaz de entender la lógica descriptiva en diversos lenguajes, o, dicho de otro modo, que las máquinas puedan describir la información de las webs y por lo tanto entiendan a los humanos de una forma eficiente (Multi, 2013).

Entender al ser humano es una tarea compleja, puesto que cada uno de ellos es un mundo diferente al igual que cada género musical, es por esta razón que en la actualidad los artistas buscan poder enlazarse con cada uno de sus fans en diferentes partes del mundo utilizando herramientas que puedan tener a su alcance, siendo las páginas web las más seleccionadas por ambas partes, cabe mencionar la problemática que surge al tener una gran cantidad de música disponible en forma ubicua que está creciendo rápidamente. Por lo tanto, existe la necesidad de técnicas de análisis para organizar dichos repositorios (Berenzweig, Logan, P.W. Ellis, & Whitman, 2003). La determinación de la similitud entre los artistas y las canciones es el núcleo de tales algoritmos, ya que proporciona una forma escalable de indexar y recomendar música.

La categoría de “compositor” como una identificación de un individuo es resultado de la división de trabajo que se ha generado históricamente en la tradición musical clásica (Marx, 1959). Por lo general cualquier ser humano tiene la capacidad biológica de crear música tomando el concepto de música no como una definición en particular sino como una agrupación de sus múltiples y coexistentes definiciones (Blacking, 1992).

Si bien es cierto, la composición requiere de distintos elementos, el principal de todos es la rima que tendrá la letra, los tipos de versos, las figuras retóricas como: la hipérbole, el símil, la sinestesia acoplándolas a diferentes estructuras como una carta, cuento, poema o incluso una analogía, es así que una canción puede ser una historia, un cuento, un juego, inclusive palabras inventadas sin sentido, por ello podemos decir que el único límite al componer una canción será únicamente la creatividad.

Los compositores anglohablantes, por la facilidad del medio en el que se desenvuelven y la cantidad de información generada en inglés, poseen mayor apertura a diferentes herramientas tecnológicas diseñadas para brindar recomendaciones dentro de su entorno de trabajo, a diferencia de los compositores hispanohablantes que deben estructurar la lírica de sus canciones por su propio medio, dando lugar en ciertas ocasiones a sentirse rechazados aun antes de dar a conocer sus temas, generando en algunos casos una posible causa de que sus letras no sean tan populares en el mundo musical.

1.2. Planteamiento del problema.

La web social es un recurso útil para quienes realizan investigaciones en informática musical. Sin embargo, no existe una forma "estándar" de integrar datos basados en la web con métodos de informática musical más comunes basados en señales (Baccigalupo & Fields, 2009).

Otro factor influyente en el ámbito musical es tener el acento correcto, como dice (Eloffson, 2013). Suecia es uno de los países europeos donde se habla mejor el inglés como segunda lengua, esto hace que muchos artistas suecos puedan escribir letras que parecen hechas por angloparlantes nativos. Para el presente caso de investigación, esto estaría estrechamente relacionado con tener correcta semántica, ya que no solo el inglés tiene sus diferentes acentos y modo de interpretar el significado de las palabras, sino que, a su vez, el español los tiene de la misma forma.

La extracción de entidades y el análisis de caracteres son tareas que pueden tener un alto grado de dificultad y resultar también costosas, estas tareas están siendo abordadas por investigadores y empresas, sin embargo, actualmente hacen falta

herramientas comerciales de semántica lírica musical eficaz. A esta situación se le une el hecho de que la mayor parte de las aproximaciones existentes se han realizado para el inglés, y hay muy pocas herramientas dedicadas al español (Casado Valverde, 2013).

En el presente documento se realiza una propuesta para la extracción de características y análisis de texto lírico musical en contenidos web mediante aprendizaje de máquinas, para brindar recomendaciones a compositores latinoamericanos, siendo una herramienta de apoyo para las compañías discográficas o de manera individual para letristas que, por falta de este tipo de herramientas, muchas veces sus composiciones no llegan a tener el impacto deseado dentro del ámbito musical.

1.3. Justificación.

En el presente trabajo de investigación, se pretende examinar la similitud basada en una fuente rica de metadatos: letras de canciones. Las letras tienen varias ventajas sobre otras formas de metadatos. En primer lugar, las transcripciones de muchas canciones populares están disponibles en línea. Así, a diferencia de otras formas de metadatos como las preferencias de los usuarios, las letras en algunos casos resultan una tarea sencilla de recopilar. Además, no son subjetivos; sólo hay una transcripción "verdadera" para una canción. Esto contrasta con formas más subjetivas de metadatos, como opiniones de expertos o transcripciones MIDI (Musical Instrument Digital Interface). Finalmente, las letras proporcionan una descripción mucho más rica de la canción que las formas simples de metadatos tales como el título, el artista y el año y discutible contener el "contenido" verdadero para muchas canciones.

En 1988, un escalador británico llamado Joe Simpson escribió un libro titulado *Touching the Void*, un relato desgarrador acerca de la muerte en los Andes peruanos. Recibió buenas críticas, pero, sólo obtuvo un éxito modesto que pronto fue olvidado. Entonces, una década más tarde, ocurrió algo extraño. Jon Krakauer escribió *Into Thin Air*, otro libro acerca de una tragedia de alpinismo en el Everest, que se convirtió en

un éxito editorial. De pronto, el libro de Simpson se comenzó a vender de manera sorprendente, a partir de la publicación de Krakauer, las ventas cada vez aumentaban, incluso pasó 14 semanas en la lista de los best sellers de la New York Times. Siendo así, las ventas superaron en más del doble que Into Thin Air. ¿Qué pasó? ¿Porque resurgió repentinamente Touching the Void? Las recomendaciones de Amazon sugerían el libro Touching the Void cada vez que un usuario compraba Into Thin Air debido a compras de usuarios anteriores. De no ser por los primeros usuarios que realizaron esta compra conjunta, Touching the Void tal vez nunca hubiese salido a la superficie. Con este caso se observa que existe una gran cantidad de contenido que puede interesar a un público, pero que no está a la mano. El mundo de la música también sufre de este tipo de experiencias, donde artistas que producen contenido de similares características a la de artistas del momento permanecen desconocidos debido a que no cuentan con la maquinaria publicitaria de artistas que pertenecen a una disquera reconocida.

Por otra parte, con el desarrollo de la herramienta de recomendación lírica como entrega final, se pretende llegar a brindar recomendaciones por género musical para que los compositores hispanohablantes tengan en sus manos la capacidad de ligar sus conocimientos con el texto semántico que se espera exponer dentro de la aplicación.

Para lograr el éxito de la herramienta, surge una serie de procesos que se llevarán a cabo para asegurar que la información recolectada está dentro del auge musical de estos tiempos, para ello, el (Application Programming Interface) API¹ seleccionado brindará el texto lírico de las mejores posiciones dentro de cada género musical. Es por esta razón que el aprendizaje de máquinas, brindará el mayor realce a la investigación, ya que aportará dentro de la selección del texto en general, con cierta ponderación, para poder clasificar por genero cada apartado musical.

¹ Conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. (Weinberger)

1.4. Objetivos.

a. Objetivo General

Extraer características y analizar texto lírico musical en contenidos web mediante aprendizaje de máquinas para brindar recomendaciones a compositores latinoamericanos.

b. Objetivos Específicos

- i. Recolectar información de texto lírico musical dentro de páginas World Wide Web (WWW) de forma metódica y automatizada por medio de motores de búsqueda, para realizar la extracción de características análogas por género musical explorando por orden de relevancia las canciones más escuchadas.
- ii. Implementar el análisis de texto, seleccionando la técnica de aprendizaje automático que reduzca el porcentaje de error en la identificación del texto analizado.
- iii. Realizar un aplicativo web distinguiendo la usabilidad con el usuario dentro del ambiente de pruebas para facilitar la interpretación de la información que contiene el prototipo.
- iv. Analizar por medio de pruebas unitarias y funcionales los resultados del aprendizaje automático insertados dentro del prototipo para estimar el porcentaje de satisfacción del usuario por medio de una encuesta descriptiva a estudiantes de la Carrera de Ingeniería en Sistemas.

1.5. Alcance.

El proyecto se llevará a cabo hasta la implementación de un prototipo que tendrá como funcionalidades las siguientes actividades:

- i. Acceso y Extracción de datos de Last.fm o similar mediante un crawler Web.
- ii. Análisis de texto lírico musical mediante modelos de aprendizaje de máquinas según el género musical y seleccionando el idioma español.

- iii. Visualización e interpretación del texto lírico musical dentro de contenidos web.
- iv. Valoración de la representación del análisis de resultados obtenidos del aprendizaje automático enlazado en la aplicación mediante pruebas funcionales que permita la visualización de satisfacción por medio de una encuesta descriptiva a un determinado grupo de población.

El prototipo será desarrollado en un ambiente web y los datos almacenados en la aplicación servirán para la realización de pruebas funcionales (figura 1), las mismas que se pretende realizar a estudiantes de la carrera de Ingeniería en Sistemas e Informática de la Universidad de las Fuerzas Armadas ESPE por la factibilidad en cuanto al acceso de usuarios que se puede tener dentro de la universidad, además, que estén próximos a egresar, puesto a que son una población que podría manejar perfectamente una página web así como interpretar los datos que se encuentren almacenados, pudiendo estos tener diferentes hobbies y como no, la interpretación musical sea uno de ellos.

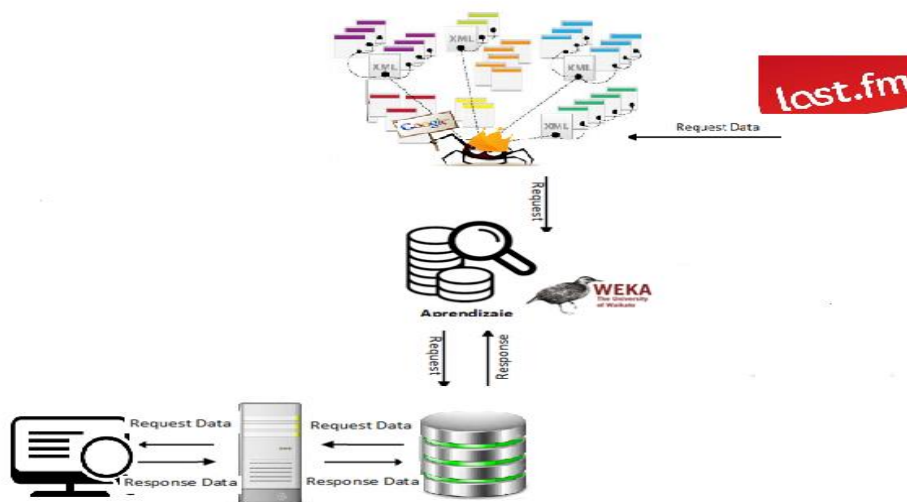


Figura 1 Arquitectura del Sistema

En la figura 2, se puede observar el modelo para la extracción de características, es decir, la letra musical de cada canción por género y autor que se buscara con la ayuda de last.fm api, es así como trabajara el crawler web que nos arrojará datos para posteriormente poder procesarlos.

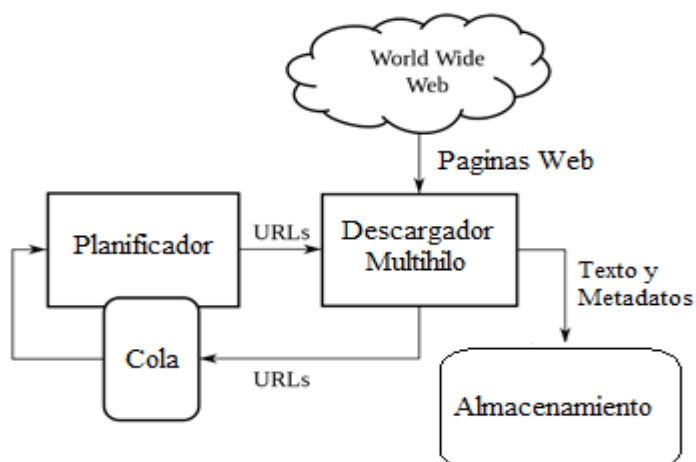


Figura 2 Arquitectura de un Web Crawler
Fuente: (Vela, 2012)

Finalmente, en la figura 3 (Sánchez-Montañés, Luis Lago, & González) se muestra el proceso de análisis de texto lírico para de esta manera poder almacenar la información en la base de datos de la aplicación.

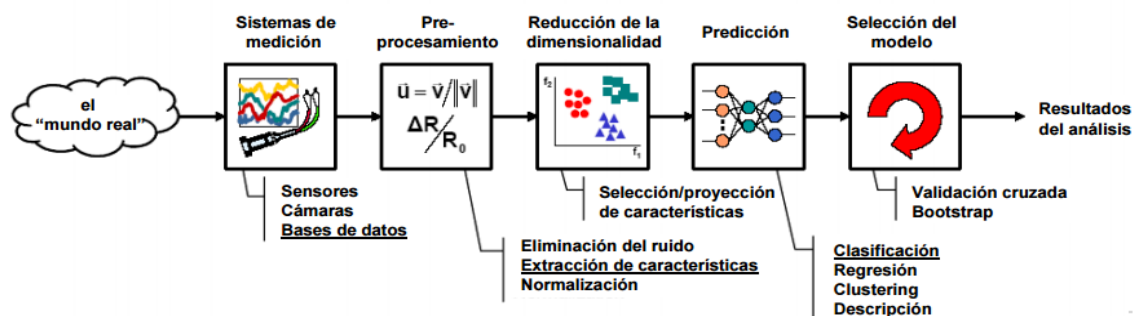


Figura 3 Proceso de análisis de texto lírico

1.6. Hipótesis

H_0 : El prototipo desarrollado dentro de la presente investigación tendrá en el periodo de pruebas un margen de efectividad en la categorización de géneros musicales de al menos 50% del total de la población analizada.

H_1 : El prototipo desarrollado dentro de la presente investigación no tendrá en el periodo de pruebas un margen de efectividad en la categorización de géneros musicales de al menos 50% del total de la población analizada.

1.7. Estructura del contenido

La presente investigación se estructura con cinco capítulos que se describen: El Capítulo I expone los antecedentes del tema de investigación a desarrollar, para obtener conocimiento de lo que existe en relación a la investigación, se expone también la problemática existente en la actualidad, de igual manera se presenta la justificación en base a investigaciones relacionadas al tema, se definen objetivos tanto general como específicos, para poder detallar el conocimiento que se desea alcanzar y definir las etapas que se requieren en el desarrollo del tema de investigación. Por último, teniendo en cuenta la magnitud del desarrollo, se especula sobre el resultado que tendrá la investigación, haciendo referencia a dos hipótesis que se espera verificar al culminar con la tesis.

Seguidamente, el Capítulo 2 abre paso al marco teórico y los trabajos desarrollados sobre temas estrechamente vinculados al tema de investigación. Se presenta entonces dentro de marco teórico a temas que sustentan el desarrollo y que son esenciales conocer para pasar al siguiente paso que sería la producción del prototipo, como son: i.) Extracción de características ii.) Análisis de texto lirico musical iii.) Inteligencia Artificial iv.) Aprendizaje Automático v.) SCRUM vi.) Weka. El Capítulo 2 finaliza exponiendo los trabajos relacionados que permiten la recolección de técnicas y herramientas que servirán como guía para el desarrollo del presente apartado.

En el Capítulo 3, se presenta la arquitectura que requiere cada proceso para llegar a feliz término con el prototipo. El apartado es desarrollado con la metodología ágil SCRUM, que empieza por el diseño del modelo de aprendizaje, seguido por la construcción y terminando con la implementación del modelo de aprendizaje incluyendo siempre al prototipo. Como este capítulo se ha denominado como Diseño, y al trabajar con Scrum permite que cada proceso exponga figuras y tablas para que sustenten el desarrollo de la investigación.

El análisis de datos se presenta el Capítulo 4, donde se requiere una cierta cantidad de datos, una vez que la aplicación ha sido manipulada por la población que

se especificó en el capítulo primero, se tiene entonces que discutir los resultados arrojados, para ello se muestra en tablas tipo pastel los porcentajes y cantidades de repuestas sobre el cuestionario propuesto al manejar el prototipo.

Para finalizar, en el Capítulo 5, se redacta las conclusiones y recomendaciones obtenidas en el desarrollo de la investigación, aquí se busca alcanzar una interpretación final de todos los datos involucrados en la investigación, así como proporcionar sugerencias de los resultados obtenidos a quienes pretendan investigar a futuro sobre los temas que se involucraron para concluir con el desarrollo del tema tratado.

CAPÍTULO 2

MARCO TEÓRICO Y TRABAJOS RELACIONADOS

2.1. Marco Teórico

A lo largo de este capítulo se pretende explicar los conceptos básicos relacionados al tema de investigación propuesto, las técnicas y tecnologías que sustentan el desarrollo y aplicación de la Inteligencia Artificial, logrando justificar por medio de artículos e investigaciones científicas el concepto del caso de estudio, los mismos que darán paso al diseño de la arquitectura del prototipo propuesto.

Primero, se realizará una explicación sobre la extracción de características, se detallará el funcionamiento de un web crawler para posteriormente mostrar el proceso de análisis de texto en un caso de estudio real, así como algunas de las técnicas utilizadas para su implementación. Considerando que la investigación está encaminada al campo de inteligencia artificial, se presentará algunas definiciones, así como las técnicas más comunes dentro del campo tecnológico, se revisará algunas las aplicaciones que están relacionadas con el aprendizaje de máquinas, y se expondrá algunos conceptos, sobre los métodos, las técnicas y los algoritmos de aprendizaje, permitiendo conocer las características que serán establecidas para el desarrollo de la aplicación.

Siendo un prototipo entregable a lo que se pretende llegar, se trabajará con una metodología de desarrollo ágil, la misma que brindará soporte para estructurar, planificar y controlar el proceso, se describe a SCRUM detallando tanto los procesos como los participantes que se involucran, por otra parte, dentro del análisis de texto se tiene como aplicativo a la herramienta (WEKA), se expondrá un concepto breve y una explicación de los entornos de trabajo que esta herramienta maneja.

Por último, se hará un recuento de los criterios que se exponen dentro de las investigaciones relacionados, en las cuales se describen las aplicaciones que sustentan los temas involucrados dentro de la investigación a desarrollar, lo que

permite conocer diferentes ámbitos de investigación que están actualmente poco perfeccionados.

2.2. Extracción de características

Según (Witten & Eibe, 2005), la extracción de características es el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Es por esta razón que, para el presente trabajo de investigación, la extracción de características es el primer proceso que se ha desarrollado, debido a que a partir de este se recolecta la información de canciones desde un sin número de páginas web.

En la actualidad la extracción de características tiene gran cantidad de aplicaciones computacionales y dentro de la industria del entrenamiento se presenta las siguientes aplicaciones:

- Composición musical.
- Predicción de los ganadores del festival de Sundance.
- HSS: Predicción del posible éxito de un grupo musical.
 - <http://www.hitsongscience.com/>
 - Norah Jones, Maroon 5. v Aquí hay una forma de ganar mucho dinero.
 - Pandora, Last-fm
 - <http://www.music-map.com/>

Como se citó anteriormente, dentro de la industria del entrenamiento, una de las aplicaciones que permite la predicción de un posible éxito musical es el Api de Last.fm, es por este motivo que, para realizar una mejor extracción de características, se ha escogido a dicha Api que se encuentra disponible en la web, puesto que, al trabajar con éxitos musicales de la actualidad, proporciona una información mucho más confiable dentro del tema de investigación que se ha desarrollado.

Además, una de las aplicaciones que mayor acogimiento que ha tenido dentro del entorno de extracción de características son los metabuscadores, ya que se encargan de localizar la información en diferentes motores de búsqueda, recorriendo

páginas web de forma sistemática y automática, además de recolectar el Uniform Resource Locator (URL) para después procesarlas.

2.2.1. Web Crawler

Un rastreador web es un programa que puede buscar y descargar documentos automáticamente desde computadoras host en redes como la World Wide Web (WWW) (Estados Unidos Patente nº US6263364 B1, 2001). Cuando un rastreador web recibe un conjunto de URL iniciales, el rastreador web descarga los documentos correspondientes, extrae cualquier URL contenida en esos documentos descargados y descarga más documentos utilizando las URL recién descubiertas. Este proceso se repite indefinidamente o hasta que se produce una condición de parada predeterminada. En la actualidad hay más de 1.000 millones de páginas web en la World Wide Web y el número está en continuo crecimiento; por lo tanto, los rastreadores web necesitan estructuras de datos eficientes para realizar un seguimiento de los documentos descargados y las direcciones descubiertas de los documentos que se descargarán.

Después de que un documento es descargado por el rastreador web, el rastreador web puede extraer y almacenar información sobre la página descargada. Por ejemplo, el rastreador web puede determinar si la página descargada contiene una nueva URL desconocida para el rastreador web, y puede encolar esas URL para su posterior procesamiento.

Los motores basados en rastreadores consideran muchos más factores que los que pueden encontrar en las páginas web. Por lo tanto, antes de poner cualquier página web en un índice, un rastreador verá cuántas otras páginas del índice están vinculando a la página web actual, el texto utilizado en los enlaces al que apunta el usuario, cuál es el rango de página de las páginas de enlace, si la página está presente en algunos directorios en categorías relacionadas, etc (Ringe, Francis, & Altaf, 2012).

Los motores de búsqueda indexan los resultados de los rastreadores web y luego realizan búsquedas cuando se consultan (Radhakishan, Farook, & Selvakumar, 2013). Por esta razón, se puede decir que en un rastreador web uno de los

componentes más relevantes que sirven para recopilar páginas web son los motores de búsqueda, en otras palabras, el rastreado web, es el medio inteligente de navegación más manejado por el motor de búsqueda. Entonces, un web crawler, no es otra cosa que el responsable de manejar y descargar la información que se presenta en la web directamente hasta el repositorio del motor de búsqueda para posteriormente poder procesarlo.

Al seleccionar un tema musical, y una vez que se ha detallado el proceso, se dará al web crawler únicamente URL's que contengan el tipo de información de acorde al tema, puesto que como menciona (Gupta & Johari, 2009), el objetivo principal de un Web crawler es proporcionar datos actualizados a un motor de búsqueda.

2.3. Análisis de texto lírico musical

A continuación, se detalla paso a paso un breve ejemplo del proceso de análisis lírico musical que fue extraído del tutorial de (Baccigalupo & Fields, 2009). El ejemplo muestra el funcionamiento de las características que brinda un web crawler como parte del procedimiento de extracción de información, cabe recalcar que dicho ejemplo sirvió como guía para dar comienzo al primer desarrollo dentro de la investigación.

2.3.1. Evaluación de hipótesis

Según los autores (Baccigalupo & Fields, 2009), para detallar la evaluación de la hipótesis se debe responder a la siguiente pregunta: ¿Cómo se evaluaría la tasa de precisión? Entonces, el proceso de evaluación tiene a los siguientes apartados que responden a la pregunta y se enumeran a continuación: i.) Crear una colección local de canciones variadas. ii.) Asignar con una etiqueta el género. iii.) Ejecutar el algoritmo del ejemplo y verificar la salida

Se detalla a continuación apartado por apartado para tener una noción del proceso a realizar. Al seleccionar una colección local de canciones variadas se refiere al tener un enfoque web, ya que para este proceso existe un sin número de datos de música que se pueden recuperar fácilmente de la web, así como los miles de canciones que se pueden descargar de forma gratuita.

Asignarlos con una etiqueta de género, quiere decir que se hace referencia a las diferentes API que poseen un contenido específico, un API web le permite recuperar datos en un formato compacto desde un sitio a través de consultas simples. Las canciones en Jamendo² por ejemplo, tienen una etiqueta de género adjunto, por lo que no tiene que decidirse por alguna de ellas, de igual manera, al trabajar con una API web automáticamente se simplifica el proceso.

Al ejecutar el algoritmo, se podrá tener un reconocedor de género específico, es decir, que una vez que se brinda las características correspondientes, el algoritmo podrá clasificar las canciones por género musical para poder distribuirlas en carpetas y así lograr que el manejo de la información sea más accesible.

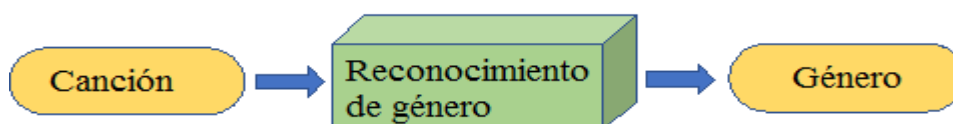


Figura 4 Proceso de reconocimiento de género musical
Fuente: (Baccigalupo & Fields, 2009).

La figura 4 describe de forma gráfica el proceso que realiza el algoritmo, para verificar el correcto funcionamiento del algoritmo, primero se debe evaluar una gran cantidad de ejemplos de etiquetas de la forma tradicional, es decir, de manera manual, luego de ello, verificar el porcentaje de aciertos para tener un estimado del error que el algoritmo puede arrojar dentro de su funcionamiento; esto debido a que en el proceso de la estimación de error intervienen varios factores en los objetos musicales tales como canciones, artistas, álbumes, listas de reproducción, usuarios, etc., se podría elegir cualquier parámetro para trabajar con el algoritmo.

2.3.2. Comparación de letras por género

² Jamendo. - Plataforma donde músicos y amantes de la música de todas partes del mundo se conectan.

La comparación de letras tiene como objetivo identificar el número de palabras que definen a un género musical, es decir, entre más letras se obtengan de la web, mayor será la cantidad de comparaciones brindando exactitud en el reconocimiento de género. Esto se deduce ya que mientras más letras se adquiera para analizar el contenido, mayor será la cantidad de palabras que se podrán comparar, logrando así definir de mejor manera el género musical de acuerdo a la repetición de las letras en diferentes canciones.

Existe una base de datos de letras de canciones en línea, una de ellas es Lyricsfly³ que proporciona una API web para recuperar letras, con detalle del API, se puede crear un script de Ruby, el mismo que se almacenara en un archivo luego de procesar la información. Como ejemplo se muestra el script `c/lyricsfly_2.rb`, el mismo que dentro del ejemplo permite especificar el nombre del artista y el título de la pista, dando como resultado la letra de canción requerida con los campos ingresados.

```
$ ruby lyricsfly_2.rb =>> Imagine there's no
"John Lennon" Imagine Heaven
It's easy if you try
No Hell below us ...
```

Figura 5 Comando de extracción de texto
Fuente: (Baccigalupo & Fields, 2009).

- Análisis basado en letras.

Las características textuales de las letras de canciones están relacionadas sin duda con el género musical que las representa, de igual manera en el ejemplo que se está tratando se presenta que las letras del género Hip-hop poseen mayor cantidad de signos de interrogación que las del género Country, este resultado fue arrojado luego de evaluar 29 canciones de Hip-hop y 41 Country, sin duda es una característica

³ Lyricsfly. - Motor de búsqueda con tecnología Google Suggest que utiliza una base de datos de más de medio millar de registros musicales.

que aunque no se prevé muestra una particularidad dentro del análisis, pudiendo encontrar estas mismas peculiaridades en otro tipo de estudios.

Al extraer una gran cantidad de información por medio de las letras de canciones se tiene la obligación de realizar el análisis con un lenguaje de programación que esté acorde a las necesidades del proceso, dentro de las características principales serían aquellos que contengan bibliotecas para recuperar páginas y analizar XML, entre los cuales se tienen a: i.) lenguajes de programación declarativos ii.) lenguajes de programación imperativos iii.) lenguajes de programación orientado a objetos, los mismo que a su vez puedan agregar datos de diferentes API web en el transcurso del análisis.

Actualmente existen aplicaciones web que permiten la integración y reutilización de diferentes características dentro del desarrollo web, la técnica más conocida es la denominada forma mash-up, la misma que puede descubrir relaciones musicales ocultas entre diferentes dominios, por esta razón muchos sitios web identifican objetos musicales a través de un conjunto específico de ID, es de esta manera como funciona Musicbrainz⁴ ya que permite hacer coincidir fácilmente el mismo elemento en varias páginas.



Figura 6 Funcionamiento de la enciclopedia MusicBrainz.
Fuente: (Baccigalupo & Fields, 2009).

2.3.3. Tendencias reveladoras

Las tendencias se vinculan a los gustos que están presentes en la actualidad y causan una inclinación de una persona hacia una cosa determinada. Es por esta razón que muchas veces una tendencia puede revelar las preferencias que una determinada

⁴ Musicbrainz. - Enciclopedia musical abierta que recopila metadatos musicales y los pone a disposición del público.

población tiene por cierto tipo de canciones, pero en muchas ocasiones, la población, únicamente es atraída por la letra de la música más que por el género al que esta pertenece, es por ello que únicamente la población escucha artistas que están de “moda” solo para estar al día de las últimas tendencias.

El uso de redes sociales en la actualidad permite tener información sobre tendencias relevantes a tan solo un clic de cualquier página en la red, cabe mencionar que otro tipo de recolector intenso de información son los Mavens⁵, que de cierta manera son los primeros en proveer de las tendencias nacientes musicales más relevantes, por esta razón el visitar ciertas API de música dará a conocer las tendencias musicales más escuchadas en la actualidad, teniendo como resultado una recomendación confiable de acorde a las necesidades del momento.

Dependiendo del API, se podrá partir de ciertos lineamientos generales, por ejemplo, sobre las tendencias del artista del mes, es por ello que surge la necesidad de vincular el análisis con un tema revelador, puesto que dará a conocer que artista o en el caso del tema de investigación desarrollado que canciones serán las más escuchadas por género musical. Se puede decir entonces que este tipo de tendencias es posible recuperar de comunidades web que están relacionas con la música como Last.fm, debido a que proporciona a cada usuario la lista de amigos y los artistas más tocados en un período determinado.



Figura 7 Tendencias por lista de amigo en redes sociales.

Fuente: (Baccigalupo & Fields, 2009).

⁵ Mavens. - Expertos en un campo específico cuya función es transmitir un conocimiento a los demás en el campo respectivo.

Para poder revelar información, el API trabaja de la siguiente manera: i.) Recupera la lista de amigos de ese usuario, ii.) Recupera los artistas más tocados por los amigos durante este y el mes anterior, imprimiendo a aquellos que han "crecido" más en este período y excluyendo a los artistas que el usuario ya conoce.

Es por esta razón que los datos sociales de la web ayudan a descubrir tendencias, y es así como los datos para las tendencias implican una dimensión temporal, un contexto (amigos, ubicación geográfica, ...) y una clase de objetos (artistas, pistas, ...) siendo esta una opción de poder observar más transparente y 'humano' en lugar de usar filtros colaborativos para futuras recomendaciones, así también, los contenedores API acortan y borran el código proporcionando una información más confiable para tratarla posteriormente.

2.3.4. Captura de datos sociales

Existe actualmente ciertas redes sociales para músicos, como por ejemplo MySpace y Soundcloud, así como diferentes sitios web para redes de músico a músico dentro de los cuales se puede otorgar acceso a la música pública de un artista, registrar relaciones entre músicos en la misma red y proporcionar datos sociales en el dominio de la música.

Los datos en las redes sociales pueden ser de gran utilidad para la informática musical, ya sea que se realice i.) un trazado de redes de amigos ii.) obtener la lista de personas de SoundCloud la que el usuario "sigue", la ventaja de las redes sociales es de conectar a una persona con el usuario inicial, un ejemplo son los músicos que se relacionan en comunidades en línea entre sí. Las redes sociales pueden extraer y trazar fácilmente, parcial o completamente la línea de conocimientos sobre un tema en particular. Los investigadores pueden beneficiarse de varias aplicaciones: generación de lista de reproducción, sistemas de recomendación entre otros.

2.3.5. Recolección de comentarios

- Evaluación subjetiva

Debido a que los investigadores necesitan comentarios en muchos escenarios, podemos mencionar entre los principales a los sistemas de recomendación, así como también al análisis basado en el estado de ánimo y la composición automática, etc. (Baccigalupo & Fields, 2009). Al tener un conocimiento leve sobre el contenido de las canciones, los autores pueden tener la certeza de que su composición esta de acorde a las tendencias musicales, logrando así potencialmente llegar a millones de usuarios.

2.4. Inteligencia Artificial

En la actualidad uno de los proyectos más ambiciosos de la informática es la inteligencia artificial, por tal motivo es difícil definir exactamente qué es y los alcances que tiene. Es de fundamental importancia conocer los orígenes de su nombre, es decir el significado de la palabra inteligencia y así mismo el de la palabra artificial, mismos que según (Arauz, 1998, p. 1) son:

- Inteligencia, es la capacidad de comprender, evocar, movilizar e integrar constructivamente lo que se ha aprendido y de utilizarlo para enfrentarse a nuevas situaciones (Zampayo, 2004, p. 10).
- Artificial, es aquel cuyo producto origen es no natural, sino que fue hecho por la mano o arte del hombre.

Ahora bien, según (Gutiérrez, 2006, p.11) la inteligencia artificial es una de las áreas más fascinantes y con más retos de las ciencias de la Computación ya que ha tomado a la inteligencia como la característica universalmente aceptada para diferenciar a los humanos de otras criaturas ya sean vivas o inanimadas, para construir programas o computadoras inteligentes. Por otro lado (Bourcier, 2003, p.56) expresa que la inteligencia artificial es una rama de la informática que intenta reproducir las funciones cognitivas humanas como el razonamiento, la memoria, el juicio o la decisión y, después, confiar una parte de esas facultades, que se consideramos signos de inteligencia, a los ordenadores. Conservando el enfoque de los autores anteriores, se puede decir que actualmente la inteligencia artificial viene siendo una disciplina de gran aporte para la ciencia y la tecnología, ya que en el transcurso del tiempo ha permitido conocer una serie de características básicas que pueden simular

el comportamiento inteligente del ser humano, dicha ciencia se ha concentrado entonces principalmente en algunas capacidades que solo el ser humano podría tener. Debido a lo expuesto anteriormente se ha visto en la necesidad de ir desarrollando infinidad de sistemas, los mismo que tiene por objetivo no solo perfeccionar el comportamiento del ser humano, sino que a su vez tienen la finalidad de reproducir sus capacidades.

De la misma manera es de vital importancia detallar las técnicas de manipulación del conocimiento, estar al tanto de cómo se dieron a conocer nuevas áreas de investigación que incluyen áreas de percepción y el lenguaje natural, cada una ligada a la visión y habla, así como la generación, la traducción, y la comprensión del lenguaje natural.

Según (Huerta, 2009, p. 18) una vertiente más de las incursiones de la Inteligencia Artificial se ha dado en el desarrollo de sistemas que ayudan a tareas de expertos, en la resolución de problemas en campos especializados (como en la realización de análisis químicos) en el campo de la ingeniería (diseño, detección de fallos, planificación de manufacturación, etc.), en el análisis científico, en la medicina, en el análisis financiero, etc.

Se puede ver como se dieron a conocer de manera más delineada las técnicas con las que se puede llegar a resolver problemas mediante la IA, según (Arauz, 1998, p. 1) una técnica de la Inteligencia Artificial es un método que utiliza conocimiento representado de tal forma que:

- Representa generalizaciones.
- Es comprendido por las personas que lo proporcionan
- Se puede modificar fácilmente.
- Puede usarse en gran cantidad de situaciones

2.4.1. Técnicas de Inteligencia Artificial

La inteligencia artificial se apoya en diferentes herramientas para la solución de problemas, dichas herramientas presentan a su vez distintas técnicas las mismas que

aportan elementos fundamentales dentro de las áreas que contiene la Inteligencia Artificial, entre algunas se tiene:

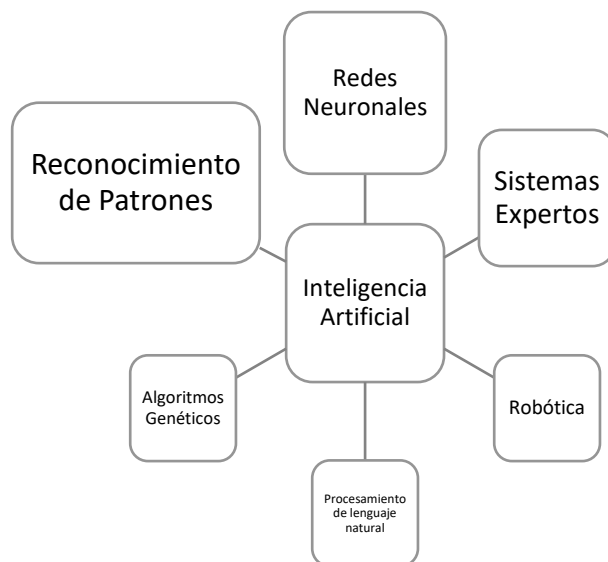


Figura 8 Técnicas de la Inteligencia Artificial

A continuación, se detalla brevemente cada una de las técnicas que se mencionó en la figura anterior.

Redes Neuronales

Según (Vásquez Padilla, 2010) son sistemas compuestos por estructuras de red con un gran número de conexiones entre diferentes capas de procesadores, los cuales a su vez tienen asignadas diferentes funciones, dentro de dichos procesadores se efectúa una labor de aprendizaje por la reproducción de las salidas de un conjunto de señales de entrenamiento.

Sistemas Expertos

Los sistemas expertos o también conocidos como sistemas basados en el conocimiento, almacenan el conocimiento de expertos en un ámbito específico y su solución se presente mediante una deducción lógica.

Por otro lado, (Vásquez, 2009, p. 12) dice que los sistemas expertos estudian la simulación de los procesos intelectuales de los expertos humanos que les permiten

interactuar con objetos del mundo real y llevar a cabo tareas de forma precisa, rápida y cómoda, semejantes a las tareas propias de un ser humano por medio del procesamiento de información y las técnicas para la solución de problemas.

Robótica

Según (Huerta, 2009, p. 29) la robótica se ocupa de tareas motrices y perceptuales, es decir la robótica es la conexión inteligente entre la percepción y la acción. La construcción de robots autónomos se realiza teniendo presente ciertas capacidades como lo son:

- La percepción básica, misma que implica la visión, la capacidad de identificar y reconocer sonidos, la habilidad de identificar olores y el sentido del tacto.
- La función motriz, comprende la habilidad de moverse en forma autónoma y la manipulación de símbolos.

Por otro lado, (Zampayo, 2004, p. 15) refiere que la robótica tiene por objetivo diseñar y desarrollar máquinas que sean capaces de realizar procesos mecánicos y manuales mediante la interacción de un sistema de control y un sistema sensorial con el que cuentan, permitiendo así, responder a los cambios que surgen en el entorno del mundo real.

Procesamiento de lenguaje natural

Según (Huerta, 2009, p. 29) el lenguaje natural, también llamado lenguaje ordinario, es el que utiliza una comunidad lingüística con el fin primario de la comunicación, y se ha construido con reglas y convenciones lingüísticas y sociales durante el período de constitución histórica de nuestra sociedad. Es decir, a través del lenguaje natural surge el fenómeno de la comunicación y por ende es una forma de transmitir el conocimiento. El procesamiento del lenguaje natural dentro de la IA consiste en:

- Procesamiento del lenguaje escrito, requiere el conocimiento léxico, sintáctico y semántico de las palabras, y del mundo real.

- Procesamiento del lenguaje real, requiere conocimientos de fonología y de la información para manejar ambigüedades que se presenten en el habla; también requiere de los conocimientos para el procesamiento de lenguaje escrito.

Se puede decir que el procesamiento del lenguaje natural es una de las técnicas más interesante en la IA, ya que tiene por objetivo estudiar el lenguaje de los seres humanos para poder acceder desde una computadora hasta todo tipo de seres inteligentes.

Algoritmos Genéticos

Según (Huerta, 2009, p. 27), un algoritmo genético normalmente trabaja sobre la representación de una posible solución a un problema dado (casi siempre cadena finita), y sobre ella se aplican operadores genéticos para combinar las bondades de las soluciones mediante la reproducción. Para medir la oportunidad de solución se crea una función de aptitud que califica a las soluciones propuestas. Se puede considerar a estos algoritmos, como un procedimiento de búsqueda y optimización, basado en mecanismos genéticos de selección natural de los seres vivos.

Reconocimiento de patrones

Según (Vázquez, 2009, pág. 12) trata de diferentes técnicas de clasificación para identificar los subgrupos con características comunes en cada grupo, y con el grado de asociación se obtiene una conclusión diferente. Los algoritmos desarrollados en esta área son herramientas útiles en otros campos como en el reconocimiento de lenguaje natural, la visión por computadora, reconocimiento de imágenes, reconocimiento de señales, el diagnóstico de fallos de equipos, el control de procesos, etcétera.

2.5. Aprendizaje automático

Cualquier cambio en un sistema que le permita funcionar mejor la segunda vez en la repetición de la misma tarea o en otra tarea extraída de la misma población (Simón, 1983). Por otra parte, se puede decir que el aprendizaje automático estudia

cómo se puede llegar a construir un programa que pueda mejorar de manera mecánica por medio de la experiencia que este adquiera.

Una vez que se ha conocido las diferentes áreas de la inteligencia artificial, se puede apreciar que la mejor manera para desarrollar técnicas que permitan aprender a las computadoras es el aprendizaje automático. El aprendizaje automático es una consecuencia natural de la intersección de la informática y la estadística, se centra en la cuestión de cómo hacer que las computadoras se programen a sí mismas a partir de la experiencia más cierta (Tom , 2006).

Entonces el aprendizaje automático es la rama de la Inteligencia Artificial, la misma que se dedica a estudiar agentes o programas que basados en su experiencia, pueden aprender e inclusive evolucionar para lograr una tarea específica cada vez de mejor manera. Para tener una idea del proceso de evaluación por el que ha pasado el aprendizaje automático se presenta a continuación la figura 9, la misma que detalla en forma breve lo mencionado anteriormente.

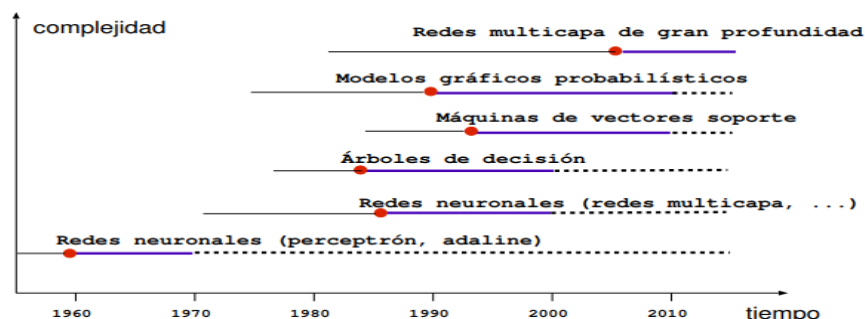


Figura 9 Evolución de la inteligencia Artificial.
Fuente: (Vidal Ruiz & Casacuberta Nolla, 2017).

Para cada tecnología, la línea continua indica el periodo de desarrollo teórico-experimental y la de puntos el periodo de vigencia como tecnología consolidada. (Vidal Ruiz & Casacuberta Nolla, 2017)

2.5.1. Métodos de Aprendizaje

Por otro parte se tiene que dentro de la rama de la inteligencia artificial algunos pilares que sostienen el correcto entendimiento de cómo poner en práctica esta

técnica, se presenta entonces ciertos algoritmos de aprendizaje que se clasifican de acuerdo a la cantidad de datos disponibles para su entrenamiento.

Para aclarar un poco la idea de manera breve se recuerda que los datos de entrenamiento son pares de entradas y salidas deseadas. En esta dimensión aparece el aprendizaje supervisado (gran cantidad de datos de entrenamiento) y no supervisado (ausencia total de datos de entrenamiento). Entre estas dos clasificaciones está el aprendizaje semi supervisado. (Demicheri & López, 2009).

Aprendizaje No Supervisado. - muchas veces llamado de auto-organización, el propio sistema trata de identificar algún tipo de regularidad en un conjunto de datos de entrada sin tener conocimiento a priori, solamente requiere de vectores de entrada para adiestrar el sistema. Esto se logra mediante el algoritmo de entrenamiento, que extrae regularidades estadísticas desde el conjunto de entrenamiento. (Cáceres Tello, 2006).

Las características principales son:

- Se pueden emplear para una descripción.
- Trata de detectar regularidades en los datos.
- Se determina por agrupaciones (clustering), agrupaciones, contornos, valores anómalos.

Mientras que su funcionamiento se base en la búsqueda principalmente de: i.) Características ii.) Regularidades iii.) Correlaciones iv.) Categorías

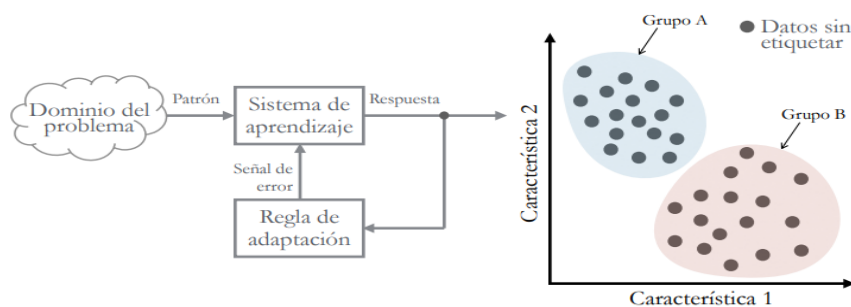


Figura 10 Modelo No Supervisado- Agrupamiento.
Fuente: (Gómez Flores).

Aprendizaje Supervisado. - el aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. (Matich, 2001).

Dentro de las características que podrían definirlo se tiene:

- Es empleado para la predicción.
- Se aprende un clasificador
- Sirve para poder explicar las causas que pueden llevar a tomar una decisión.

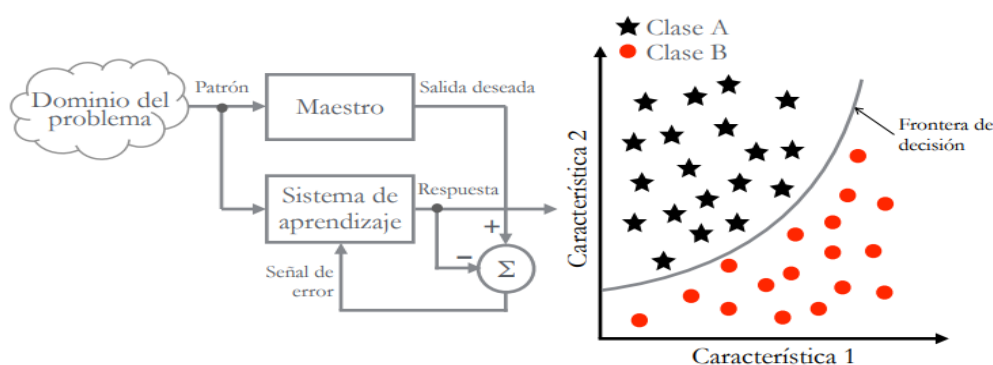


Figura 11 Modelos Supervisado: Clasificación.
Fuente: (Gómez Flores).

Aprendizaje Semi Supervisado. - es una combinación del aprendizaje supervisado y no supervisado. Puesto que asignar etiquetas o clases a los datos puede ser muy costoso, se puede recurrir a la opción de usar a la vez un conjunto de datos etiquetados de tamaño pequeño y un conjunto más extenso de datos no etiquetados, mejorando así la construcción de los modelos. Esta técnica es la usada por el aprendizaje semi-supervisado. En este método, se ha de tener en cuenta que no siempre los datos no etiquetados son de ayuda al proceso de aprendizaje. Por lo general, se asume que los datos no etiquetados siguen la misma distribución que los etiquetados para que el uso de datos sin etiquetar sea útil. (Gallardo Campos, 2009)

En la actualidad existen un sin número de métodos de aprendizaje semi-supervisado, los mismo que ofrecen ciertas características determinadas. Una manera

de saber escoger el método que resulta más adecuado es observando cuál de todos ellos se podría ajustar mejor a las necesidades de cierto problema específico.

2.5.2. Algoritmos y métodos de aprendizaje

A través de la investigación sobre el comportamiento de los métodos de aprendizaje, se ha desarrollado algunos algoritmos con la finalidad de obtener predicciones más precisas y a su vez reducir los tiempos de entrenamiento, dentro de una encuesta en KD Nuggets se obtiene a los 10 mejores algoritmos y métodos mencionando a clustering en segundo lugar, sin embargo, enlistarlos en un orden específico dependerá de varios factores de análisis como el tipo y el número de data, es por esta razón que todos y cada uno de ellos tienen sus fortalezas y debilidades dependiendo el campo para el que son estudiados, es decir, la diferencia fundamental entre estos algoritmos radica en la forma en que se almacena el conocimiento. Así pues, un breve ejemplo es que dentro las redes neuronales, el conocimiento se traduce en una serie de pesos y umbrales que poseen las neuronas, mientras que en K-Means se basa en la agrupación mediante clústeres.

2.5.3. Aplicaciones del aprendizaje automático

El aprendizaje automático no solo tiene como finalidad crear nuevas aplicaciones, sino que además permite mejorar las aplicaciones ya existentes para que de cierta manera puedan complementarse ayudando a perfeccionar las funcionalidades que estas tienen, como el aprendizaje automático es una rama descendiente de la estadística y de la informática, son numerosas las aplicaciones dentro de estas áreas, así como en las ramas de la industria y la ciencia.

En el área de la informática, el aprendizaje automático está relacionado con aplicaciones tan heterogéneas como el desarrollo de robots humanoides o Internet. Un ejemplo de estas aplicaciones es el PageRank⁶ usado por Google, que ampara

⁶ Algoritmo de Google cuya función es medir la importancia y la calidad de una página web

una familia de algoritmos utilizados para asignar de forma numérica la relevancia de los documentos (o páginas web) indexados por un motor de búsqueda (Gallardo Campos, 2009).

Dentro del campo de la estadística, se pretende obtener un análisis de los conjuntos de datos, aquí se presentan los sistemas de recuperación de la información que buscan confeccionar rankings personalizados de acuerdo a la experiencia de los usuarios. Existe un sin número de campos que aplican los métodos de aprendizaje automático, no solo la rama de la estadística o la informática, actualmente, la psicología, la medicina, la robótica, la industria bancaria, industria musical entre otros emplean esta técnica para desarrollar nuevas aplicaciones.

El aprendizaje automático es de vital importancia para la clasificación de secuencias de ADN, asistir un diagnóstico a partir de la historia clínica de un paciente, reconocimientos de tumores, reconocimiento de escritura, reconocimientos de imágenes, por el mismo camino se encuentra el análisis sintáctico y morfológico de los textos.

Una de las características que diferencia a este método, es la facilidad con la que se adapta para perfeccionar un entorno automáticamente, es por esta razón que dentro de un ambiente que varía constantemente se puede presentar como solución dentro del estudio de mercados, detección de fraudes, evaluar el riesgo crediticio, etc. En sistemas de recomendación, el aprendizaje automático puede proporcionar un agrupamiento semántico, extracción de información, clasificación automática de documentos, análisis de sentimientos, entre otros. En la tabla se presenta algunas aplicaciones dentro de los campos mencionados anteriormente.

Tabla 1
Aplicaciones del aprendizaje automático

Dominio del problema	Aplicación	Información
Bioinformática	Análisis de secuencia	ADN/ Secuencias de proteínas
Medicina	Diagnóstico médico	Estudios de enfermedades, diagnósticos médicos

CONTINÚA



Economía	Detección de fraudes	Informes de facturas
Reconocimiento biométrico	Identificación de personas	Cara, iris, huellas dactilares
Clasificación de documentos	Búsqueda en Internet	Documento de texto
Análisis de imágenes	Lectura automática para ciegos	Documentos de imagen
Robótica	Reproducir el comportamiento humano	Sensores de audio, video, infrarrojos, etc.
Informática	PageRank	Páginas web e información de acceso

Fuente: (Gallardo Campos, 2009).

2.6. SCRUM

Scrum es un framework de desarrollo ágil de software. El trabajo es estructurado en ciclos de trabajo llamados sprints, iteraciones de trabajo con una duración típica de dos a cuatro semanas. Durante cada sprint, los equipos eligen de una lista de requerimientos de cliente priorizados, llamados historias de usuarios, para que las características que sean desarrolladas primero sean las de mayor valor para el cliente. Al final de cada sprint, se entrega un producto potencialmente lanzable / distribuible / comerciable. (Scrum Alliance, s.f.)

Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos (Schwaber & Sutherland, 2013). Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que se puedan mejorar. Al ser una metodología ágil, scrum se basa (Itzcoalt Alvarez) en los siguientes principios:

- Respuesta al cambio
- Desarrollo incremental con entregas frecuentes de funcionalidad
- Simplicidad, únicamente los artefactos que sean necesarios
- Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

2.6.1. Proceso Scrum

Según (Álvarez García, de las Heras del Dedo, & Lasa Gómez, 2012) Scrum define dos etapas en su proceso de organización del trabajo. La primera denominada El Sprint 0 o etapa inicial; y la segunda de iteraciones sucesivas, también llamadas Sprints.

Un Sprint es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un Sprint el producto es diseñado, codificado y probado. Y su arquitectura y diseño evolucionan durante el desarrollo (Procesos de Software, s.f.).

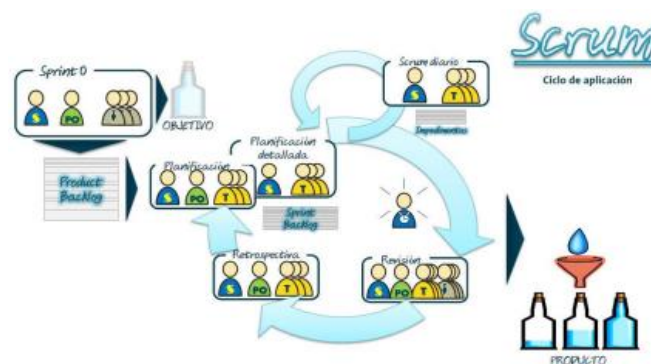


Figura 12 Ciclo de aplicación SCRUM.

Fuente: (Álvarez García, de las Heras del Dedo, & Lasa Gómez, 2012).

2.6.2. Roles

Las actividades que plantea el Scrum, guían a la elaboración del producto que el cliente necesita en base a sus requerimientos. En Scrum, el equipo de desarrollo se focaliza en construir un software de calidad. La gestión de una investigación basada en Scrum se centra en definir previamente cuáles son las características que debe tener el producto que se va a construir (qué construir, qué no y en qué orden).

a. Product Owner

Es la única persona, dentro del Scrum Team, que conoce el giro del negocio del cliente, mantiene la visión del producto, trabaja con el cliente en el levantamiento

de los requerimientos y facilita la comunicación con el equipo de desarrollo. Es también el encargado de mantener el Product Backlog actualizado y se asegura que el equipo de desarrollo construya el producto correcto.

b. Scrum Master

Esta persona debe dominar el marco de trabajo, que Scrum propone, con el fin de mantener en funcionamiento el proceso, ayudar al Product Owner con la elaboración y actualización del Product Backlog, además de gestionar los impedimentos que se van presentando en el desarrollo del proyecto.

c. Equipo de desarrollo

Es un grupo de profesionales que deben tener las habilidades necesarias para elaborar y entregar el producto en base a los requerimientos planteados por el cliente. El equipo de desarrollo tiene la autoridad para auto-organizar su trabajo con el fin de alcanzar los objetivos de cada Sprint.

2.6.3. Aplicación

Según (Scrum-Alliance, 2014) Con el fin de mantener el control y realizar el seguimiento del trabajo, que el equipo de desarrollo va a realizar, Scrum define una serie de artefactos y herramientas

a) Product Backlog

Es una lista de ideas o requerimientos priorizados del producto, de donde se desprende el trabajo que el Scrum Team debe realizar. El encargado de elaborar y actualizar este listado es el Product Owner.

b) Sprint Backlog

Es el elemento resultante de la etapa de planificación del Sprint o Sprint Planning. Contiene el número de requerimientos refinados que el equipo de desarrollo se compromete a desarrollar en el período de duración de un Sprint.

c) Product Increment

Es el artefacto más importante de Scrum ya que constituye el incremento de producto u objetivo que cada Sprint debe producir. Debe cumplir con los requerimientos planteados, ser utilizable por parte del usuario y aceptado por el Product Owner.

2.7. WEKA

Un desarrollo emocionante y potencialmente de gran alcance en la ciencia de la computación es la invención y la aplicación de métodos de aprendizaje automático. Estos permiten que un programa de computadora analice automáticamente un gran conjunto de datos y decida qué información es más relevante. Esta información cristalizada se puede usar para hacer predicciones automáticamente o para ayudar a las personas a tomar decisiones de forma más rápida y precisa (The University of Waikato, s.f.).

Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Los algoritmos se pueden aplicar directamente a un conjunto de datos o llamar desde su propio código Java. Weka contiene herramientas para el preprocesamiento de datos, clasificación, regresión, clustering, reglas de asociación y visualización. También es adecuado para desarrollar nuevos esquemas de aprendizaje automático (The University of Waikato, s.f.).

Antes de comenzar con la aplicación de las técnicas de WEKA a los datos de un dominio en particular, es muy conveniente hacer una consideración acerca de los objetivos perseguidos en el análisis. Un paso previo a la búsqueda de relaciones y modelos subyacentes en los datos ha de ser la comprensión del dominio de aplicación y establecer una idea clara acerca de los objetivos del usuario final. De esta manera, el proceso de análisis de datos, permitirá dirigir la búsqueda y hacer refinamientos, con una interpretación adecuada de los resultados generados. Al conocer el dominio que tendrá la aplicación, se ve la factibilidad que brindará Weka al realizar el análisis de datos, puesto a que una vez concentrada la información en un solo medio de

almacenamiento, se pretende posteriormente visualizar de manera que pueda ser manipulada en el prototipo en desarrollo.

Los objetivos, utilidad, aplicaciones, etc., del análisis efectuado no "emergen" de los datos, sino que deben ser considerados con detenimiento como primer paso del estudio. En esta investigación, uno de los objetivos perseguidos es relacionar los resultados obtenidos dentro del proceso de extracción de datos por medio de crawler, con características de canciones seleccionadas por género, si bien la descripción que se encuentra disponible no llegara a proporcionar información oportuna habrá que atenerse a lo que por el momento pueda ser útil. Se verá que muchas veces dentro del resultado alcanzado se puede encontrar relaciones triviales o conocidas previamente, también puede ocurrir que el hecho de no encontrar relaciones significativas sea muy relevante.

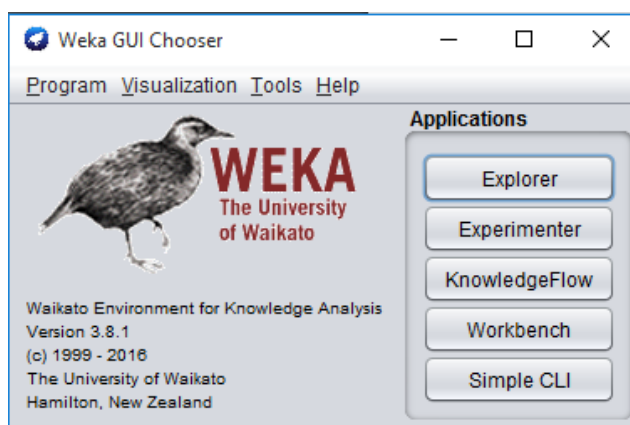


Figura 13 Interfaz de Herramienta Weka

Como se puede ver en la parte inferior de la Figura 13, Weka define 4 entornos de trabajo

- Simple CLI: Entorno consola para invocar directamente con java a los paquetes de Weka.
- Explorer: Entorno visual que ofrece una interfaz gráfica para el uso de los paquetes.
- Experimenter: Entorno centrado en la automatización de tareas de manera que se facilite la realización de experimentos a gran escala.

- KnowledgeFlow: Permite generar proyectos de minería de datos mediante la generación de flujos de información.

2.8. Trabajos Relacionados

Actualmente existen trabajos que exponen algunos criterios en base a los temas que esta investigación abarca, como es a bien conocer, se recolecta información de varios autores para que a partir de las conclusiones de sus investigaciones se pueda obtener una inclinación acerca de los lineamientos por lo que se encaminará el desarrollo del prototipo.

Como menciona (Mayer, Neumayer , & Rauber, 2008) las letras de canciones exhiben propiedades específicas diferentes de los textos tradicionales, es por esta razón que una clasificación por género musical conlleva a considerar el estilo y la rima para el procesamiento de letras, dentro de su investigación aplican la concatenación simple, además muestran el planteamiento de mejorar el procesamiento de letras por heurística para la detección de dichas características, por otra parte, dentro de la investigación de (Knees, Schedl, & Widmer, 2005) mencionan la problemática al comparar letras de múltiples fuentes, así mismo dentro de su proceso, el cual consiste en recopilar letras de internet, alinear las letras y producir una cadena de salida de la información musical, recomiendan alinear una relación del resultado con la secuencia de coincidencia más alta para clasificar la información por género musical.

Dentro de la propuesta de (Chakrabarti, Van den Berg, & Dom, 1999) introducen el concepto de crawlers focalizados, los mismos que han sido utilizados actualmente en el rastreo de la web para problemas específicos, y es que como mencionan (Rui, Fen, & Zhongzhi, 2008), la característica de este tipo de crawler es que no necesitan visitar todas las páginas web seleccionadas, sino que se enfocan solo en las URL's más relevantes, dentro de los crawlers focalizados, (Camargo Sarmiento & Ordóñez Salinas, 2013) mencionan de la existencia de técnicas y modelos de clasificación bayesiana, modelos de asociación semántica o la mezcla de diferentes modelos como la clasificación Bayesiana con el algoritmos de lógica difusa que fue profundizada en la investigación de (Qiang, 2007), además (Ibrahim, Selamat,

& Selamat, 2008) al realizar su estudio propuesto sobre crawlers dirigidos a redes sociales, propone utilizar agentes multicrawlers, por esta razón se obtiene la iniciativa de crear crawlers sociales, dentro de la investigación de (Ting, Hui-Ju, & Pei-Shan, 2009) concluyen que un crawler para redes sociales debería estar construido con la técnica de “minería de estructuras Web” debido a que la comprensión de la estructura de los sitios podría mejorar la forma en que los crawlers sociales se construyen.

Se presenta las tendencias que actualmente manejan los webs crawlers, por ejemplo, (Tadapak, Suebchua, & Rungsawang, 2010) proponen un crawler para un idioma específico (Thai) con métodos supervisados. (Shaojie, Tianrui, Hong, Yan, & Jing, 2010) proponen una mejora al algoritmo de PageRank utilizando medidas de similaridad bajo el nombre de SimRank, (Qureshi, Younus, & Rojas, 2010) diseñan un crawler llamado “visionerBOT”.

Dentro de la investigación (Álvarez Díaz, 2007) menciona que es posible diseñar soluciones crawling que van dirigidas a la extracción de la información, formula una arquitectura para sustentar a un crawler dirigido, además, sugiere que se debe considerar la información “basura” a la que está expuesta el crawler una vez que maneja los datos que se encuentran en la web, proponiendo técnicas basadas en contenido, no obstante, (Yuvarani, Iyengar, & Kannan) propone a los rastreadores enfocados, que dentro de su investigación lo denominan LSCrawler, este tipo de crawler se basa principalmente en modelos probabilísticos para predecir la relevancia de los documentos, dicho sistema explota la semántica de las palabras clave dentro del enlace, por otro lado (Boldi, Codenotti, Santini, & Vigna) proponen un crawler mucho más elaborado, UbiCrawler es un rastreador distribuido y escalable, que solventa inconvenientes con el manejo de conjunto de datos extremadamente grandes. Hablar de otras herramientas como menciona (Abello, Pardalos, & Resend, 2002) es considerar a Mercator, desarrollado completamente en Java⁷ ya que consta de ciertas características como: i.) distribuido ii.) escalable iii.) alto rendimiento, así

⁷ Lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90

mismo (Benítez Andrades, 2010) dentro de su comparación con algunas herramientas, confirma que Mercator consumió menos recursos además de lograr descargar más páginas por segundo, otra herramienta para el análisis de texto a gran escala es WebFountain desarrollado en lenguaje C++, según (Edwards, McCurley, & Tomlin) al ser un rastreador incremental, posee la capacidad de actualizar el repositorio tan pronto se rastrea la página web real, entonces, el área de investigación principal con la que está estrechamente relacionada el crawling, es sin duda aquella afín con los motores de búsqueda, que realizan mediante diferentes técnicas una extracción de datos que van de acuerdo a la necesidad para la que fue desarrollada.

La investigación de (Cadavid Rengifo & Gómez Perdomo, 2009) contribuye a la implementación de técnicas de lenguaje natural no supervisado, y se comprueba que cuanto mayor sea el volumen del cuerpo de información recolectada, más fácil será identificar los elementos válidos del lenguaje, estas técnicas normalmente trabajan con un enorme volumen de información textual que se encuentra acumulada en la web. Por otro lado (Soto & Jiménez, 2011) aplica la técnica de aprendizaje supervisado con reglas euclidianas para mejorar el análisis de la información, demostrando la factibilidad técnica con su aplicación a una base de datos para obtener la clasificación de la información, pero (Valdiviezo, Santos, & Boticario, 2010) concluyen que para ofrecer un sistema de recomendación es mejor basarse en la aplicación de técnicas de aprendizaje no supervisado, seleccionando a la técnica de clustering para encontrar características similares dentro del análisis de datos.

Por otra parte, existen varias investigaciones centradas en algoritmos de rastreo de páginas web, donde el objetivo de los algoritmos únicamente es reunir tantas páginas como sea posible enfocándose tan solo en el primer rastreo, por esta razón, (Rungsawang & Angkawattanawit, 2004) plantean dentro de su investigación, la propuesta de un algoritmo capaz de aprender de acuerdo a las bases de conocimiento que se irán construyendo gradualmente de la información de cada página web visitada desde el intento anterior de rastreo, dando al rastreador la posibilidad de aprender de la experiencia del proceso anterior, consiguiendo que éste pueda acercarse más rápido a las páginas web relevantes. (Valdiviezo, Santos, & Boticario, 2010) aportan que dentro de las experimentaciones que manejaron con la

técnica de clustering, el modelo que arroja mejores resultados dentro del análisis de datos es SimpleKmean, por otra parte, (Bragado, 2016) describe el proceso de aprendizaje no supervisado dentro de un crawler, al algoritmo de clusterización de tipo jerárquico aglomerativo para solventar las necesidades de la investigación al desarrollar un motor de búsqueda.

Un instrumento para introducir a las personas en el aprendizaje automático, es Weka, donde (Garner) indica que es una herramienta útil e incluso esencial dentro del análisis de conjunto de datos, de igual manera (Witten I. , y otros, 1999) afirman que Weka es un paso significativo en la transferencia de la tecnología de aprendizaje automático a un ambiente de trabajo de escritorio, además el entorno de trabajo de Java proporciona un soporte automático para la documentación puesto que brinda una compilación automática con HTML, (Hornik, Buchta, & Zeileis, 2008) menciona a otro de los principales entornos de código abierto disponibles para el aprendizaje automático que ha surgido de las comunidades estadísticas como es R, que al reunirse con Weka fusionan algunas de las características de las capas habitual de código de R con la interfaz tacto-visual de Weka, pero que presenta inconvenientes como i.) exhaustiva privacidad en la información, ii.) demasiada manipulación de datos iii.) una sola vía de comunicación, que para una cantidad de datos poco extensa es muy probable que funcione de mejor manera, por otro lado (Sharma, Bajpai, & Litoriya , 2012) presentan una comparación entre los diversos algoritmos de agrupamiento de Weka, sostiene que Weka es la herramienta más simple para la clasificación de datos de distintos tipos, y a su confirma que el algoritmo de clustering K-means es el algoritmo más simple que esta herramienta proporciona para principiantes en el aspecto de análisis de datos.

Por otra parte, se ha investigado temas relacionados con la extracción y análisis de texto lirico musical que brinden recomendaciones en idioma español, pero lamentablemente no hubo éxito en la búsqueda de aplicaciones de este tipo, por esta razón se sustenta entonces la problemática planteada dentro del Capítulo 1, ya que actualmente existen sistemas de recomendación únicamente para el idioma inglés.

A continuación, se presenta en la Tabla 2, las características más relevantes que sustentan las técnicas seleccionadas para el desarrollo del prototipo.

Tabla 2

Técnicas del proceso de desarrollo

Características	Aprendizaje Supervisado	Aprendizaje No Supervisado	WebCrawler	UbiCrawler
Algoritmos	K vecinos más cercanos	K medias		
Herramientas	Weka	Weka	Java, Python, C++	Java
Técnicas	Arboles de decisión	Clustering	Focalizado, social	Focalizado
Métodos	Estadísticos	No relacionales	Clasificación	Clasificación
Organización	Variables estáticas	Variables aleatorias	Agrega resultado de varios motores de búsqueda	

Dadas las características indicadas en la Tabla 2, se discierne que el prototipo se basará en una arquitectura centralizada mediante un crawler ya que la información que se pretende manejar no llega a ser tan extensa para justificar la distribución en el proceso de extracción, puesto que el tema de investigación recopila URL únicamente de características musicales y además se basa en un problema específico, un crawler focalizado se acopla a las necesidades.

Con respecto al proceso de análisis de texto, se lo realizará con la herramienta Weka, dado que cuenta con algunos entornos de trabajo que vienen inmiscuidos dentro de la aplicación, cabe recalcar que el proceso de análisis se basará entonces en el método de aprendizaje no supervisado, con la técnica de clustering y el algoritmo k-means.

2.9. Conclusiones

Dentro de las aplicaciones que se pueden manejar con Inteligencia Artificial existe muy pocas dentro del campo de clasificación de documentos.

Existe un sin número de herramientas en diferente lenguaje para el modelado y procesamiento de datos, sin duda Weka es una de las herramientas que presenta una interfaz gráfica que permite la facilidad para dentro del manejo de datos.

Scrum al ser una metodología ágil permite agilizar el proceso de desarrollo puesto que se puede dividir el concepto del problema en pequeñas tareas, en este caso, particionar la aplicación en diferentes procesos, además permite tener mayor flexibilidad a los cambios que se pueden presentar en cada una de las fases del desarrollo del software.

Dentro de los trabajos que se relacionan con el tema de investigación existen algunos en los que se habla del manejo de diferentes tipos de crawler, cada uno se basa en la extensión de características de cada tema y dependiendo de ello la selección de búsqueda con la que será configurado el crawler.

CAPITULO 3

DISEÑO

A lo largo de este capítulo se pretende dar a conocer el diseño y la arquitectura que tiene el prototipo de la presente investigación, es por esta razón que para dar comienzo al desarrollo, se ha seleccionado la arquitectura que satisface las necesidades del prototipo, al escoger entre diferentes modelos surge la necesidad de presentar una breve justificación de cada selección, como se conoce, el prototipo consta de una serie de procesos consecutivos, los mismos que para el posterior tratamiento se les denomina módulos, y dentro de la investigación se reconocieron tres módulos para llegar a feliz término.

Seguidamente, se procede al desarrollo del prototipo en base a la metodología de desarrollo ágil SCRUM, la misma que presenta un sprint por cada módulo, se detalla brevemente las tareas de cada sprint que no es más que los lineamientos en los cuales se basa la metodología de desarrollo como son: i) planificación ii) análisis iii) diseño iv) construcción v) pruebas (Orjuela Duarte & Rojas, 2008). Lo que diferencia a SCRUM de las demás es la elaboración del backlog del producto, que no es más que un listado de actividades enumeradas de acuerdo a la prioridad en base a la etapa de planificación de toda la investigación de manera general.

Por otro lado, cada sprint posee la fase de pruebas, pero en este caso la construcción del prototipo demanda de actividades incrementales evolutivas y para lograr visualizar las funcionalidades por completo del prototipo es mejor tener el producto final terminado, por este motivo se presenta la necesidad de que la fase de pruebas sea parte del siguiente capítulo de la investigación.

3.1. Arquitectura

Para la implementación del prototipo se ha utilizado un modelo N capas, las mismas que fueron distribuidas tal como se presentó de forma general dentro del capítulo 1, a continuación, se detalla los métodos y modelos que fueron elegidos para

el desarrollo de cada módulo, presentando la arquitectura y justificando la elección de cada una de ellas.

La mayoría de sistemas de recolección de información se basa en dos tipos de arquitecturas: la centralizada y la distribuida, ambas son las arquitecturas más eficaces para la búsqueda de la información que existe en los repositorios Web (Camargo Sarmiento & Ordóñez Salinas, 2013), cabe mencionar que del producto de las visitas a una página web depende la calidad y fiabilidad de los resultados de la búsqueda de un tema en particular, es por esta razón que para el desarrollo del prototipo de la investigación, la arquitectura que más se acopla a las necesidades es la arquitectura centralizada (ver figura 14), puesto que el tema que abarca el producto se enfoca en un solo caso en particular y la semilla del buscador ya tiene sus propias limitaciones.

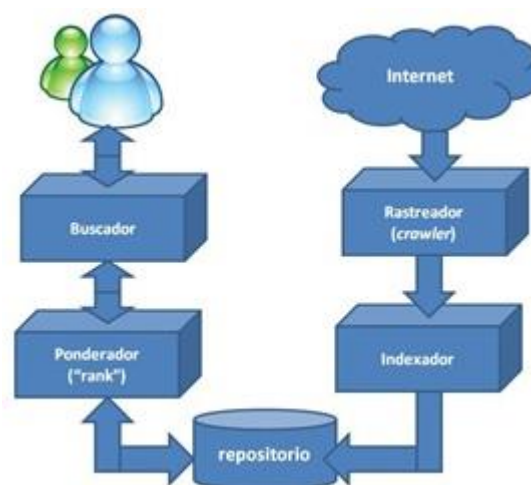


Figura 14 Arquitectura de un buscador centralizado.
Fuente: (Camargo Sarmiento & Ordóñez Salinas, 2013).

Por otro lado, como se mencionó en el análisis de la tabla 2 en el capítulo anterior, la investigación se basa en la implementación de un web crawler focalizado, puesto que es un tipo de crawler que recibe uno o varios parámetros de entrada (como frases o palabras) y rastrea la web para localizar sitios con contenido relevante a dichos términos (Camargo Sarmiento & Ordóñez Salinas, 2013). Una de las características más importantes de este tipo de crawlers es que no se dedican a coleccionar todas las páginas web visitadas, sino que únicamente se enfocan en el

conjunto de URL predefinidos con tópicos seleccionados justo antes de empezar el rastreo. Se muestra a continuación en la figura 15 la arquitectura que posee el crawler desarrollado del presente trabajo de investigación.

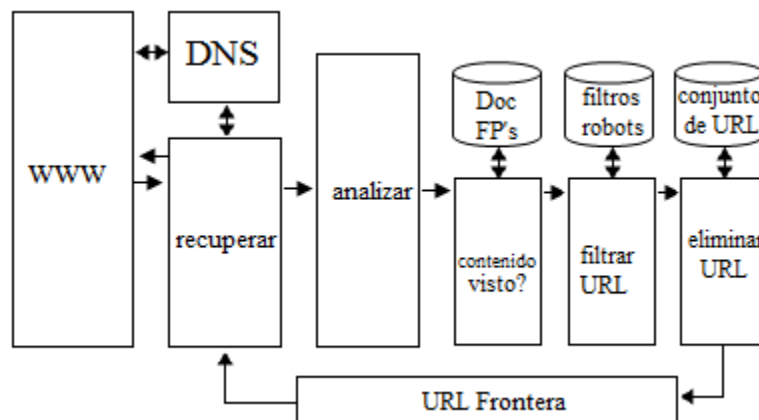


Figura 15 Arquitectura Web Crawler

Para el siguiente proceso que consiste en el análisis de toda la información previamente recolectada, y de acuerdo a las necesidades de la investigación se ha seleccionado a la herramienta Weka ya que esta contiene características que brinda confiabilidad en el manejo de información, además de permitir la visualización y la selección de datos dentro de la ejecución, una vez que se establece la tarea, Weka brinda la posibilidad de acceder a una gama de técnicas para el modelado y el procesamiento de datos, acoplándose a la selección de la técnica de clustering, la misma que consiste en adjuntar una ponderación para el agrupamiento de texto, una vez que finaliza la actividad, la información es guardada en la base de datos que gestionará el prototipo del sistema propuesto. Brevemente se muestra en la figura la arquitectura del proceso anteriormente detallado.

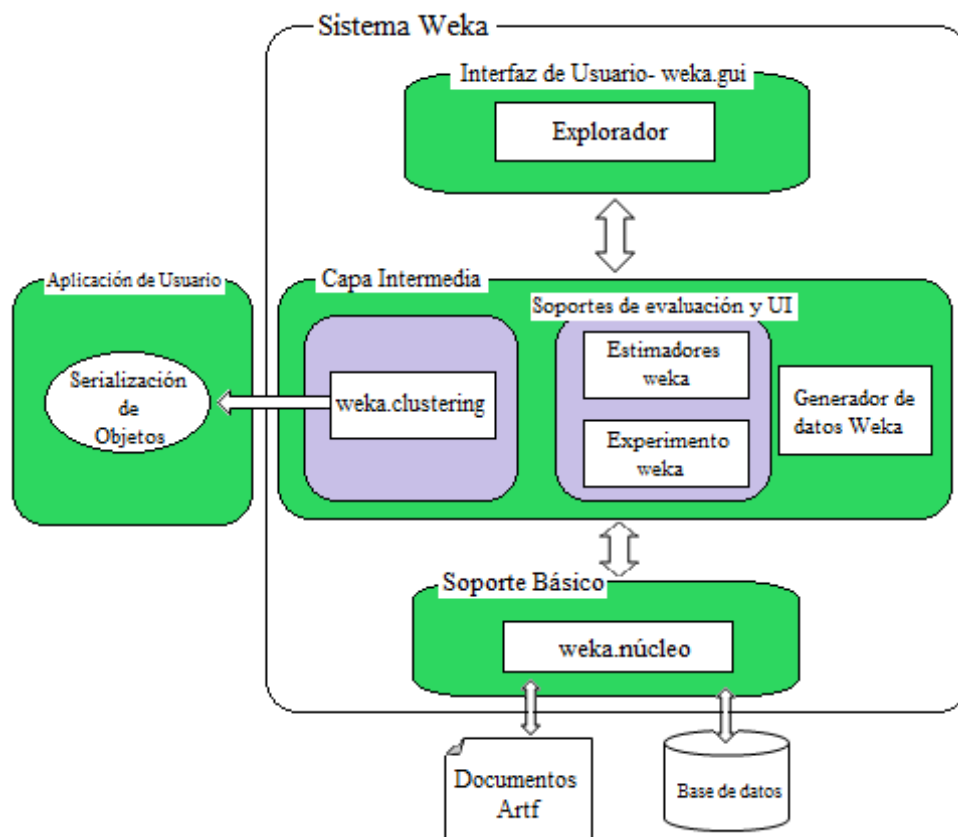


Figura 16 Arquitectura Weka.
Fuente: (Inacap).

Al desarrollar una aplicación web, se tiene por ventaja la factibilidad y la portabilidad del producto, es decir, es posible realizar una página web que esté al alcance de cualquier usuario, por esta razón dentro del último proceso de la investigación se muestra la implementación del aplicativo web. En la figura 17 se muestra la arquitectura que tendrá la página web, la misma que desplegará toda la información que fue extraída y analizada en los procesos anteriores.

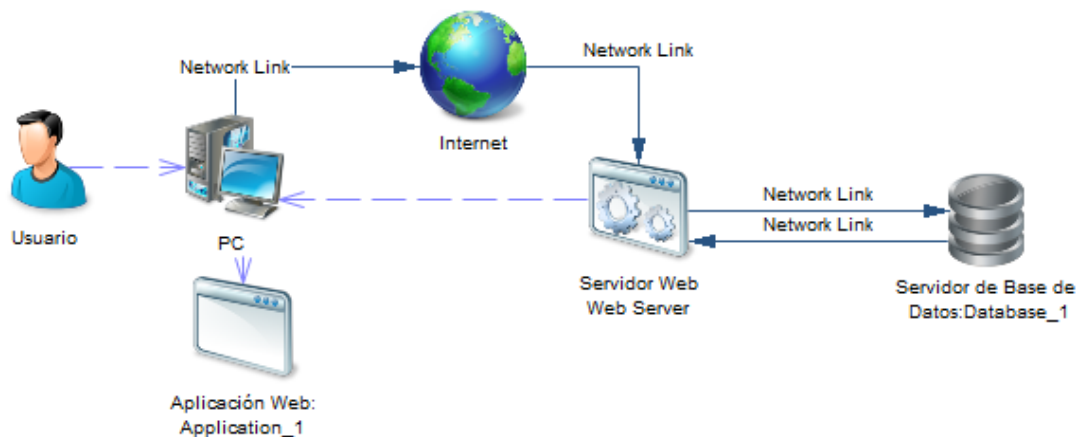


Figura 17 Arquitectura Web

3.2. SCRUM

Scrum es una de las metodologías de desarrollo ágil que pretende particionar a todo el producto en fracciones entregables llamados sprints, la ventaja de esta partición, es la distribución del tiempo de trabajo en cada uno de ellos, de igual manera cabe mencionar que estos sprints son incrementales, ya que mientras cada sprints esté terminado se avanza al siguiente hasta llegar al término del producto (Trigas Gallego & Domingo Troncho, 2012), es por estas razones que en un ambiente tan inestable como es aquel en el que se desenvuelve el software actualmente, este tipo de metodologías encajan bien para lograr un buen desarrollo del producto, puesto a que se reduce el tiempo en el que se entrega el producto pero esto no disminuye la calidad del mismo.

3.2.1. Planificación

La investigación pretende implementar una serie de procesos que llevan a mostrar los datos que se recolecta en cada uno de ellos dentro de una página web, el desarrollo de todo el proceso fue guiado en una metodología ágil la misma que tiene como punto de partida la planificación de todo el trabajo que conlleva a la realización del producto, es por esta razón que se muestra de forma global el conjunto de sprints que la investigación necesita (ver figura 18), cabe mencionar que el detalle de cada sprint, se detallará posteriormente.

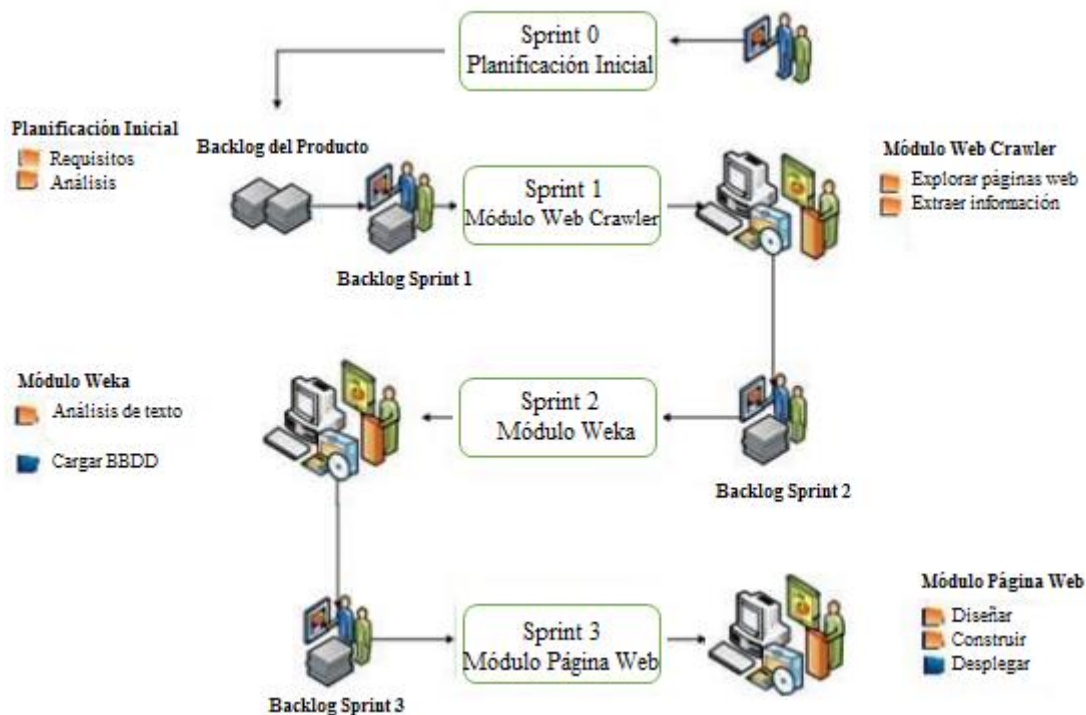


Figura 18 Planificación del proyecto

Backlog del producto

A continuación, se muestra de manera general el diagrama de procesos que pretende dar solución al tema de investigación (ver figura 19), empezando por el módulo de extracción de la información, seguidamente el módulo de análisis de la información y para terminar se presenta el módulo del desarrollo de la página web, el mismo que busca implantar la información previamente recolectada en los módulos anteriormente mencionados.

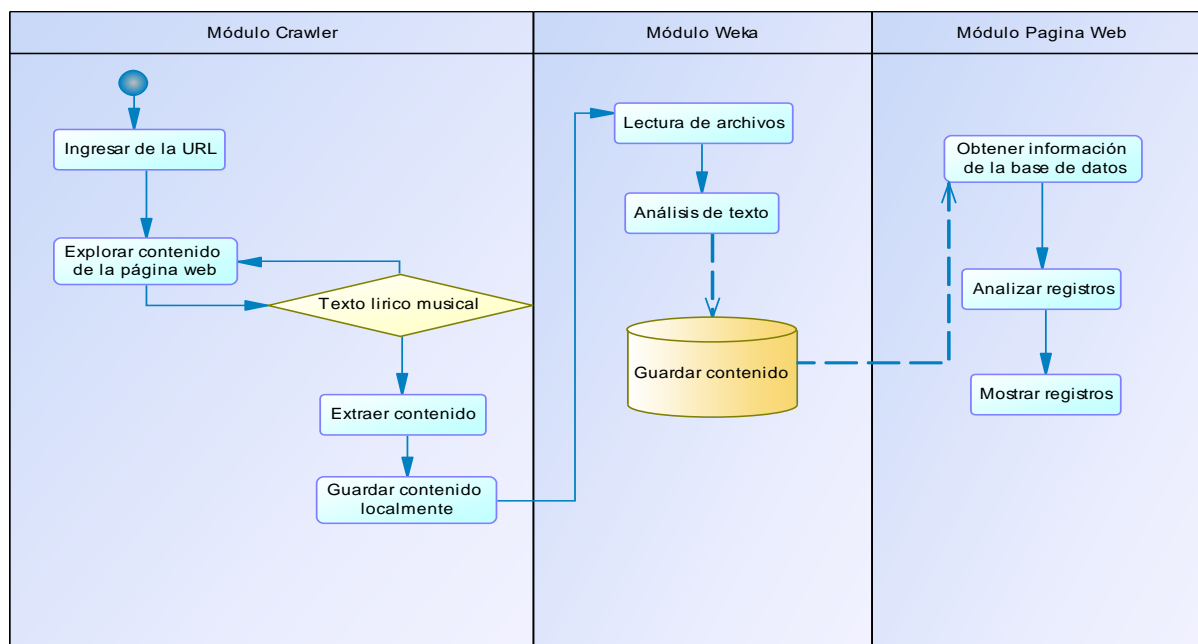


Figura 19 Diagrama de procesos

La tabla 3 muestra las prioridades del sistema en base a una ponderación numérica con respecto a la dificultad e importancia dentro de la investigación para cada módulo y sprint, dicho en otras palabras, el requerimiento más importante para la investigación será el que debe ser desarrollado primero y así sucesivamente.

Tabla3

Definición de prioridades

Número	Dificultad	Prioridad
1	Fácil	Baja
2	Moderada	Media
3	Compleja	Alta
4	Muy Compleja	Muy Alta

Para cada requerimiento la prioridad fue determinada en base a dos aspectos i) Por parte de las necesidades de la investigación ii) Por parte del desarrollador de la investigación, al final de la planificación se llegó a un consenso previo al análisis de cada uno de ellos y a la importancia numérica que se presentó con anterioridad, los resultados del consenso, se muestran en cada historia de usuario.

En base a la metodología de desarrollo Scrum, se presenta dentro de la tabla 4 a continuación el Backlog del producto, el mismo que será desarrollado en base a los ítems que se muestran a continuación, cabe mencionar que dicho Backlog se encuentra basado en el estudio de procesos de los requerimientos identificados durante la planificación del proyecto.

Tabla4

Backlog del producto

ID	Nombre	Importancia	Tiempo estimado (semanas)	Comentarios
1	Módulo Web Crawler	Alta	3	Presente en la funcionalidad de búsqueda y extracción de información
2	Módulo Weka	Alta	3	Presente dentro de la funcionalidad de análisis de la información
3	Módulo Aplicación Web	Media	1	Presente dentro de la funcionalidad que permite el retorno de información requerida

3.2.2. Sprint 1: Web Crawler

El desarrollo del siguiente sprint pretende dar solución al módulo web crawler, el mismo que es el encargado de obtener la información relacionada con el texto lírico musical que se encuentra alojado en el api de last.fm, y que a su vez descarga los archivos en forma local, seguidamente se detalla cada fase que llevó a cumplir con el objetivo del presente sprint.

Planificación

Dentro de la fase de planificación se pretende dar a conocer de manera breve los requerimientos funcionales que corresponden al módulo del web crawler (ver tabla 5 y 6), los mismos que fueron ordenados de acuerdo a la prioridad en el

procedimiento de la investigación, cabe mencionar que, al ser el primer proceso, se tiene una consideración dentro de los tiempos de desarrollo.

Tabla5

Requerimiento funcional 1

Id. Requerimiento	REQ01 Búsqueda y almacenamiento de datos
Descripción	Permite buscar y almacenar los datos que estén relacionados con texto lírico musical dentro del api las.fm
Entradas	URL del listado de canciones del api
Salidas	Contenido de las páginas web relacionadas con temas musicales dentro del URL enviado
Proceso	El web crawler busca el contenido en las páginas web que estén relacionadas dentro de la URL ingresada, luego las analiza y procede a guardar aquellas que se encuentren ligadas a la URL inicial.
Precondición/es	Conectividad a Internet
Postcondición/es	Se actualiza el almacenamiento de la información dentro del crawler.
Efectos Colaterales	El web crawler correrá únicamente cada inicio de mes.
Prioridad	Alta

Tabla6

Requerimiento funcional 2

Id. Requerimiento	REQ02 Extracción y almacenamiento de archivos
Descripción	Permite extraer y almacenar los archivos que estén relacionados con texto lírico musical dentro del api las.fm
Entradas	URL del listado de canciones del api

CONTINÚA



Salidas	Archivos de las páginas web relacionadas con temas musicales dentro del URL enviado
Proceso	El web crawler extrae el contenido de las páginas web que estén relacionadas dentro de la URL ingresada, procede a guardar localmente los archivos que contengan dicha información.
Precondición/es	Conectividad a Internet
Postcondición/es	Se procede a descargar los archivos en un directorio local distribuido por carpetas.
Efectos Colaterales	Los archivos se actualizan al inicio de cada mes, esperando la ejecución del web crawler para extraer la información.
Prioridad	Alta

Como la investigación se basa de acuerdo a los lineamientos de la metodología Scrum, y una vez que se han identificado con anterioridad las funcionalidades que tendrá el web crawler, se pueden identificar de manera clara las historias de usuario que tendrá el presente sprint, de esta forma se detalla a continuación dentro de la tabla 7.

Tabla7

Historia de usuario sprint 1

ID	Historia usuario	de	Importancia para el propietario del producto	Importancia técnica	Descripción
1	Definir la URL que contenga información de las canciones previamente seleccionadas.	URL	5	4	Se necesita realizar el filtro del idioma español dentro de las páginas web
2	Desarrollar el web crawler.	web crawler.	4	5	Se obtiene el código fuente que permite extraer la información requerida de las páginas web.
3	Descargar archivos	los archivos	3	4	Como el módulo requiere de archivos

CONTINÚA 

planos, se aceptan únicamente los links de contenido musical con letras de canciones para su posterior análisis

Análisis

Una vez que se han identificado las historias de usuario se puede apreciar claramente la línea que debe seguir el desarrollo del web crawler dando como resultado una correcta diagramación de los casos de uso que arrojó el proceso de planificación, para poder proceder la fase de implementación del Sprint 1.

- Diagramas de casos de uso Sprint 1

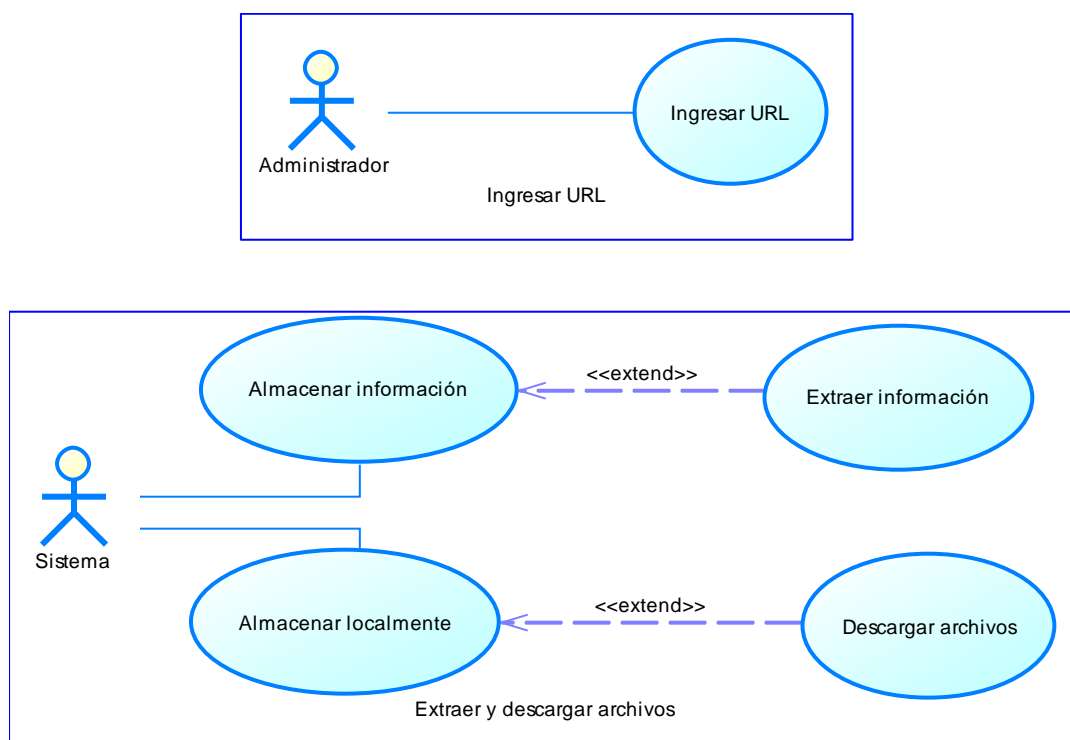


Figura 20 Casos de Uso-Módulo Web Crawler

- Especificación de casos de uso del Sprint 1

Tabla8*Especificación de caso de uso: seleccionar url*

ID	ReF-01
Descripción	Seleccionar la URL del api para el funcionamiento del Web Crawler
Precondición	Listado de canciones más escuchadas
Postcondición	
Flujo Normal	<ol style="list-style-type: none"> 1. Revisar la lista de las 100 canciones más escuchadas. 2. Comprobar si el URL seleccionado es válido.
Flujo Alternativo	
Excepciones	<p>Si la URL no cumple con las necesidades se borra de la lista.</p> <p>Si la URL ha cambiado, se toma la más actual.</p>

Tabla9*Especificación de caso de uso- extraer información*

ID	ReF-02
Descripción	Extraer la información de las páginas web que estén relacionadas con los tops 100 de música.
Precondición	ReF-01
Postcondición	Actualización de información
Flujo Normal	<ol style="list-style-type: none"> 1. Explorar las páginas web y sus respectivos links asociados. 2. Analizar similitud en el contenido de las páginas web visitadas. 3. Guardar información.
Flujo Alternativo	Si la URL analizada ya contiene las letras de canciones, se procede al siguiente requerimiento.
Excepciones	Si la URL no contiene información que esté relacionada con temas musicales no se procede al análisis.

Tabla10
Especificación de caso de uso-descargar archivo

ID	ReF-03
Descripción	Descargar el archivo y almacenar localmente en la computadora
Precondición	Análisis de contenido dentro de la URL ingresada
Postcondición	Actualización de carpetas que se guardan localmente.
Flujo Normal	<ol style="list-style-type: none"> 1. Explorar el link que contiene letras musicales. 2. Descargar el archivo. 3. Guardar localmente el archivo.
Flujo Alternativo	
Excepciones	Si el archivo no contiene la información relacionada con letras de canciones, se descarta el almacenamiento.

Diseño

- Modelo de datos

Dentro del análisis que se obtuvo anteriormente de los casos de uso, surge la necesidad de emplear dentro de la investigación un modelo de datos no relacional, puesto que no es necesario diseñar los schemas o su estructura por adelantado, a continuación, se pretende detallar de manera breve y clara en la figura 21.

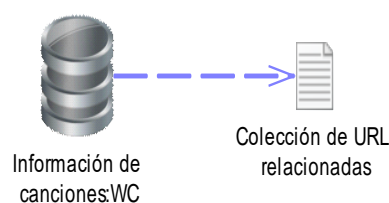


Figura 21 Modelo de Datos

Construcción

- Backlog del Sprint

Las historias de usuario que fueron descritas con anterioridad son las mismas que serán implementadas dentro del primer sprint, siguiendo los lineamientos de la metodología SCRUM es necesario definir un Backlog del sprint 1 que se muestra en la tabla 11, dicho de otra manera, se mostrará el orden de las funcionalidades específicas que fueron seleccionadas en la etapa de análisis. Se muestra a continuación las actividades que se lleva a cabo en el sprint 1:

Tabla11

Backlog sprint 1-módulo web crawler

ID Historia	Núm. Actividad	Nombre de la actividad	Asignación	Estimado (horas)
1		Codificar el Web Crawler		
	1	Codificar el web crawler para la extracción de información	Andrea Reimundo	25
	2	Almacenar páginas web relacionadas	Andrea Reimundo	10
	3	Pruebas funcionales	Estudiantes	2
2		Descargar archivos		
	1	Almacenar de manera local los archivos	Andrea Reimundo	8
	2	Pruebas funcionales	Estudiantes	2

Pruebas

Las pruebas como se puede visualizar en el backlog del presente sprint fueron realizadas a estudiantes de la Universidad de las Fuerzas Armadas “ESPE” las mismas dependieron tanto del módulo como de los componentes que lo conforman, es por esta razón que, debido a la amplitud de las mismas, éstas han sido descritas en el siguiente capítulo de la investigación.

3.2.3. Sprint 2: Módulo Weka

Dentro del desarrollo del segundo sprint se encuentra el módulo Weka, denominado de esta manera debido a que la herramienta que se empleará lleva el mismo nombre, la misma que aportará en el proceso de análisis de toda la información obtenida previamente con la ayuda de la codificación del web crawler, es por esta razón que el siguiente sprint se encarga de la lectura, así como del almacenamiento en la base de datos.

Planificación

La planificación de este Sprint se basa en los lineamientos extraídos dentro del Sprint 0, de ahí se procede a extraer los requerimientos tanto funcionales como no funcionales (ver tabla 12), los mismos que fueron considerados como parte del proceso de análisis de texto, cabe mencionar que el texto fue recopilado y descargado dentro del sprint 1. Scrum al ser un proceso incremental evolutivo, se va añadiendo funcionalidades al sistema en cada sprint.

Tabla12

Requerimiento funcional 3

Id. Requerimiento	REQ03 Leer archivos y analizar información
Descripción	Permite leer y analizar el texto de los archivos descargados para poder almacenar en la base de datos.
Entradas	Archivos planos
Salidas	Contenido de los archivos almacenados en la base de datos.
Proceso	La lectura y análisis de texto se basa en el uso de la herramienta Weka, la misma que permite obtener los datos agrupados por género musical para almacenar en la base de datos.
Precondición/es	Debe existir archivos planos en el directorio analizado.
Postcondición/es	Almacenamiento de la información dentro de la base de datos del sistema.

CONTINÚA



Efectos Colaterales	La base de datos podrá ser actualizada cada inicio de mes.
Prioridad	Alta

Se procede entonces a realizar las historias de usuario (ver tabla 13), las mismas que contienen las actividades de todo el proceso, el cual consiste en el análisis de texto previamente obtenido del módulo anterior, puesto que dicho análisis requiere del manejo de gran cantidad de datos, se ha escogido la herramienta Weka, en la cual se obtuvo una correcta ponderación en base a la facilidad de agrupamiento con el algoritmo previamente seleccionado.

Tabla13

Historia de usuario sprint 2

ID	Historia usuario	de	Importancia para el propietario del producto	Importancia técnica	Descripción
1	Leer los archivos.		5	5	Se procede a la apertura de cada uno de los archivos que se encuentra en las carpetas almacenadas localmente.
2	Analizar datos.	los	5	4	Se analiza archivo por archivo de cada carpeta, se agrega una ponderación respectiva al agrupamiento de la información.
3	Almacenar información en la base de datos.		2	4	Guarda los datos que fueron analizados dentro de la base de datos del sistema para su próxima gestión.

Análisis

Una vez que se han identificado las historias de usuario se puede apreciar claramente la línea que debe seguir el módulo Weka, dando como resultado una correcta diagramación de los casos de uso que arrojó el proceso de planificación, para poder proceder con la fase de implementación del Sprint 2.

- Diagrama de casos de uso Sprint 2

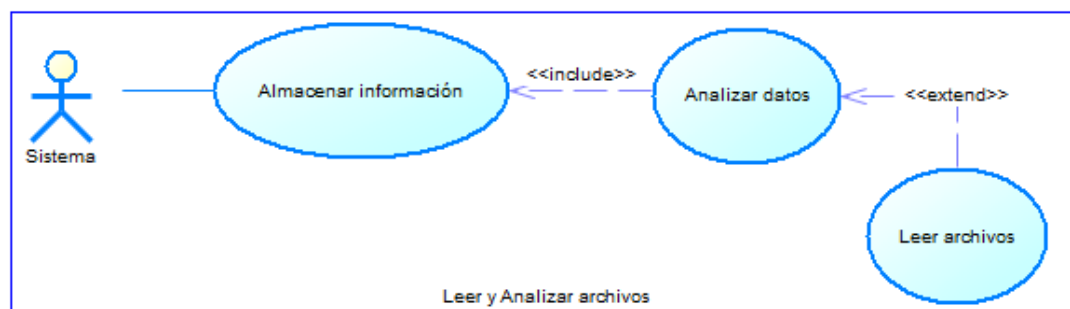


Figura 22 Casos de Uso-Módulo Weka

- Especificación del Caso de Uso del Sprint 2

Tabla14

Especificación de caso de uso-leer y analizar archivos planos

ID	ReF-04
Descripción	Leer y analizar los archivos que fueron descargados para el almacenamiento dentro de la base de datos.
Precondición	ReF-03
Postcondición	Se actualizará la base de datos del sistema
Flujo Normal	<ol style="list-style-type: none"> 1. Leer los archivos descargados. 2. Analizar el contenido. 3. Almacenar datos en la base del sistema.
Excepciones	Lectura de archivos planos que tengan el mismo formato inicial.

Diseño

- Modelado de datos

Dentro del análisis que se obtuvo anteriormente de los casos de uso, surge la necesidad de emplear dentro de la investigación un modelo de datos no relacional, puesto que no es necesario diseñar los schemas o su estructura por adelantado, a continuación, se pretende detallar de manera breve y clara en la figura 23.

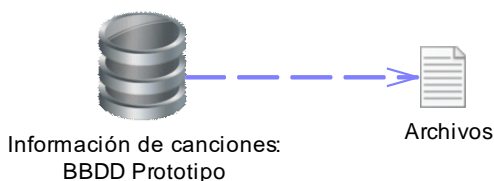


Figura 23 Modelos de Datos

Construcción

Las historias de usuario que fueron descritas con anterioridad son las mismas que serán implementadas dentro del segundo sprint, siguiendo los lineamientos de la metodología SCRUM es necesario definir un Backlog del sprint 2 en la tabla 15, dicho de otra manera, se mostrará el orden de funcionalidades específicas que fueron seleccionadas en la etapa de análisis, se muestra a continuación las actividades que se lleva a cabo en el sprint 2:

Tabla15

Backlog sprint2: módulo weka

ID Historia	Núm. Actividad	Nombre de la actividad	Asignación	Estimado (horas)
1		Utilizar la herramienta Weka		
	1	Procesamiento de datos con el modelo previamente seleccionado	Andrea Reimundo	25
	2	Almacenar información en la base de datos del sistema.	Andrea Reimundo	8
	3	Pruebas funcionales	Estudiantes	2

Pruebas

Las pruebas como se puede visualizar en el backlog del presente sprint fueron realizadas a estudiantes de la Universidad de las Fuerzas Armadas “ESPE” las mismas que dependieron tanto del módulo como de los componentes que lo conforman, es por esta razón que, debido a la amplitud de las mismas, éstas han sido descritas en el siguiente capítulo de la investigación.

3.2.4. Sprint 3: Módulo Página Web

El desarrollo del siguiente sprint pretende dar solución al módulo página web, este módulo es el encargado de mostrar al usuario la información obtenida en los procesos anteriores, dentro de diseño y elaboración de la página web se consideró que los datos deben estar distribuidos dentro del cuerpo, de la cabecera y el pie, dicho de otra manera, las palabras que brindan recomendaciones por género musical se agruparon en una misma columna, para cumplir con los objetivos de este nuevo sprint se tuvo las consideraciones que a continuación se detallan.

Planificación

La planificación de este Sprint se basa en los lineamientos extraídos dentro del Sprint 0, de ahí se procede a extraer los requerimientos tanto funcionales como no funcionales (ver tabla 16), los mismos fueron considerados como parte del proceso diseño y elaboración de la página web, cabe mencionar que la información fue recopilada y procesada dentro de los sprint 1 y 2. Scrum al ser un proceso incremental evolutivo va añadiendo funcionalidades al sistema en cada sprint, en este caso, al ser el proceso final este sprint es el encargado de la visualización del producto terminado.

Tabla16

Requerimiento funcional 4

Id. Requerimiento	REQ04 Diseñar y elaborar la página web
Descripción	Permite visualizar la información recopilada en los procesos anteriores.
Entradas	Registros de la base de datos.
Salidas	Información desplegada por género musical.

Proceso	Esbozar dentro de la página web el posicionamiento colectivo de la información.
Precondición/es	Debe existir información en la base de datos del sistema.
Postcondición/es	Datos correctamente distribuidos para la visualización del usuario.
Efectos Colaterales	Ninguno.
Prioridad	Alta

Se procede a realizar las historias de usuario pertenecientes a este módulo (ver tabla 17) las mismas que contienen las actividades de todo el proceso, el cual consiste en el diseño y elaboración de la página web, cabe mencionar que la información que se despliega ha sido previamente analizada para satisfacer las necesidades del usuario, es por esta razón que el agrupamiento de los datos se debe distribuir correctamente para que la exploración de la página web sea simple e intuitiva.

Tabla17

Historia de usuario sprint 3

ID	Historia usuario	de	Importancia para el propietario del producto	Importancia técnica	Descripción
1	Diseñar la página web.		5	4	Se procede a seleccionar un tipo de página web de acuerdo a las necesidades del sistema.
2	Elaborar página web.	la	5	3	En base al diseño se codifica la página web en el lenguaje de programación seleccionado.
3	Visualizar página web.	la	5	4	Se observa los datos que contiene la base de datos del sistema de manera distribuida para una correcta interpretación del usuario.

Análisis

Una vez que se han identificado las historias de usuario se puede apreciar claramente la línea que debe seguir el módulo de la página web, dando como resultado una correcta diagramación de los casos de uso que arrojó el proceso de planificación, para poder proceder con la fase de implementación del Sprint 3.

- Casos de uso Sprint 3

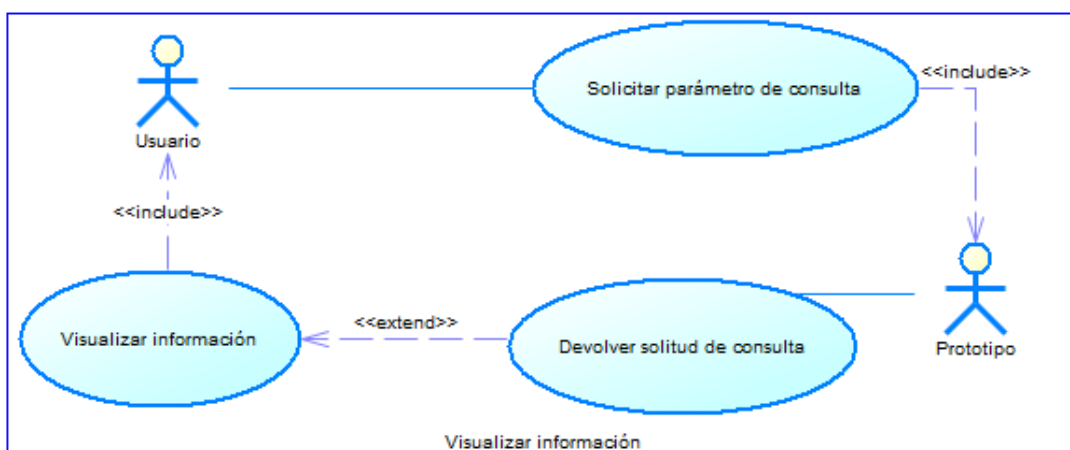


Figura 24 Casos de Uso-Módulo Página Web

- Especificación del caso de uso Sprint 3

Tabla18

Especificación de caso de uso-diseñar y elaborar página web

ID	ReF-05
Descripción	Diseñar y elaborar la página web que despliega la información almacenada en la base de datos del sistema.
Precondición	ReF-04
Postcondición	Se retorna la información solicitada
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario solicita una recomendación que es tomada como parámetro de consulta hacia el prototipo. 2. El prototipo devuelve la respuesta dentro de la distribución de la información en la página web. 3. El usuario visualiza la recomendación solicitada.

Flujo Alternativo

Excepciones

No encontrar el género que el usuario consulta.

Diseño

- Modelado de página web

Dentro del análisis que se obtuvo anteriormente de los casos de uso, se presenta la necesidad de diseñar un bosquejo de lo que será la página web para poder proceder a su codificación, es por esta razón que para este Sprint a diferencia de los anteriores el modelo que se presenta en la figura 25, es únicamente de cómo se verá visualmente la página web.

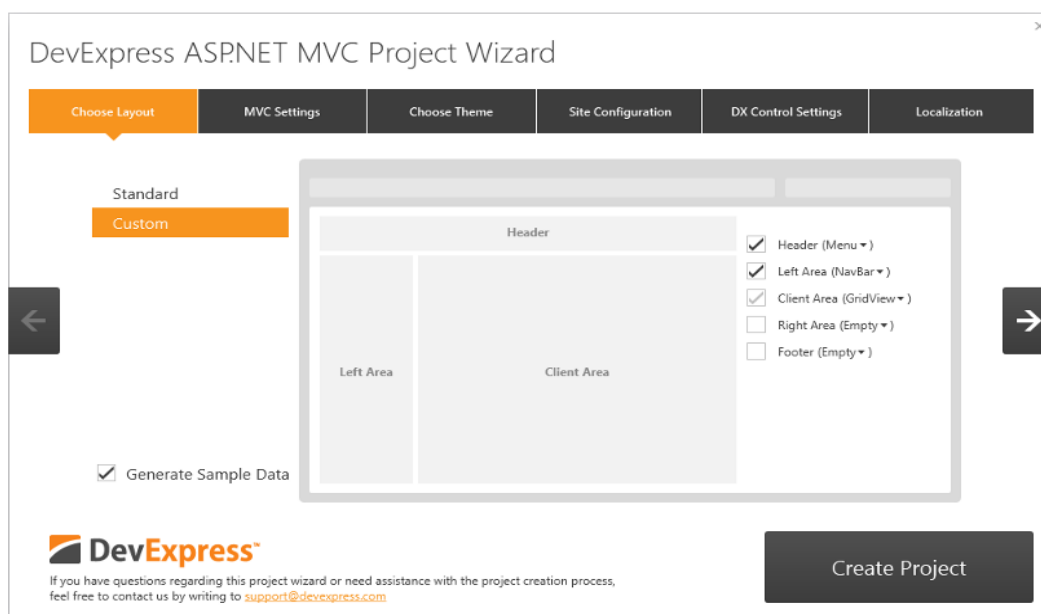


Figura 25 Diseño Página Web

Construcción

Las historias de usuario que fueron descritas con anterioridad son las mismas que serán implementadas dentro del último sprint, siguiendo los lineamientos de la metodología SCRUM es necesario definir un Backlog del sprint 3 en la tabla 19, dicho de otra manera, se mostrará el orden de funcionalidades específicas que fueron

seleccionadas en la etapa de análisis, se muestra a continuación las actividades que se lleva a cabo en el sprint 3:

Tabla19

Backlog sprint 3-módulo página web

ID Historia	Núm. Actividad	Nombre de la actividad	Asignación	Estimado (horas)
1		Codificar la página web		
	1	Diseño de la página web.	Andrea Reimundo	10
	2	Elaboración de la página web.	Andrea Reimundo	8
	3	Pruebas funcionales	Estudiantes	2

Pruebas


Las pruebas como se puede visualizar en el backlog del presente sprint fueron realizadas a estudiantes de la Universidad de las Fuerzas Armadas “ESPE” las mismas que dependieron tanto del módulo como de los componentes que lo conforman, es por esta razón que, debido a la amplitud de las mismas, éstas han sido descritas en el siguiente capítulo de la investigación.

3.3. Maquetado del sistema

Una vez que se estudió el esqueleto al que se le dio forma dentro de la planificación del sprint dos, se procede a la descripción de la estructura que tendrá el sistema de forma general, puesto que dentro de las necesidades que representa el desarrollo se considera el diseño de una página web dinámica cuyo objetivo es plasmar los datos recolectados en un solo pantallazo, dicho de otra manera, la navegación será entre la página principal.

Sistema de Reomendación Musical Latinoamericana

Salsa Baladas Folklore Rock Pop Cumbia




La salsa fue consolidada como un éxito comercial por músicos de origen caribeño (cubanos, puertorriqueños y dominicanos) en la ciudad de Nueva York en la década de 1960, si bien sus raíces se remontan a décadas anteriores en países del Gran Caribe. 3 La salsa abarca varios estilos como la salsa dura, la salsa romántica y la timba

Orden	Palabra	Repetición
1	Deseo	50
2	Luna	48
3	Amandote	47
4	Fantasia	30
5	Hechizo	25

Figura 26 Maquetación del sistema- Género Salsa

Sistema de Reomendación Musical Latinoamericana

Salsa Baladas Folklore Rock Pop Cumbia



La balada es un género musical aparecido en la década de 1960 que alcanzó gran popularidad en los países de habla hispana y portuguesa de América Latina y España. Se caracteriza por ser una canción interpretada en tiempo lento, siempre sobre temas de amor.

Orden	Palabra	Repetición
6	Perdedor	55
7	En la noche	40
8	Eterno	20
9	Cielo	15
10	Arena	2

Figura 27 Maquetación del sistema-Género Baladas

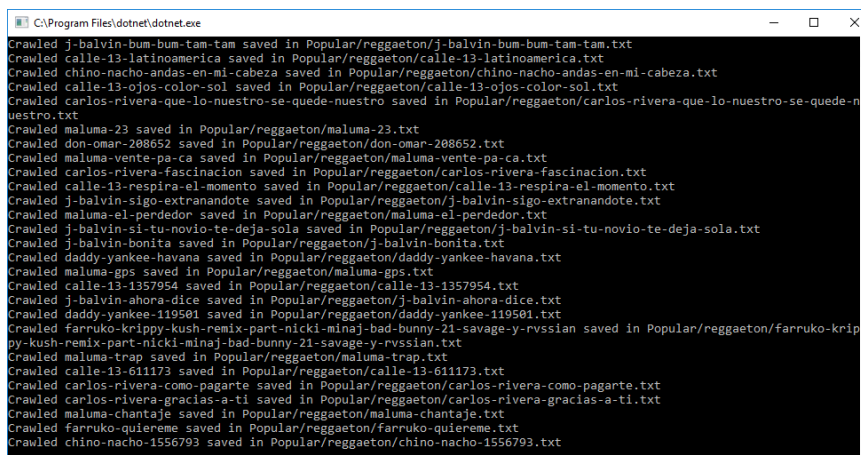
Para el maquetado del mismo se manejó la herramienta Axure Pro 8, la misma que permitió lograr una mejor comprensión del entorno en el cual se va a desenvolver el sistema una vez que entre en producción, al terminar este proceso, se obtuvo los ambientes que se reflejan en las figuras 26 y figura 27, cabe mencionar que se tendrá un entorno diferente por cada género musical, recalando que cada uno de ellos contendrá el mismo diseño pero con la tabla de recomendación de acorde a cada género musical.

3.4. Desarrollo de software

Debido a que la investigación se basó en los lineamientos de SCRUM cada sprint terminado contiene una parte de todo el proceso, es por esta razón que este punto se enfoca al agrupamiento del código fuente dentro del desarrollo, con el objetivo de visualizar punto por punto la importancia que cada uno tuvo para que el software final tuviera las funcionalidades especificadas en la etapa de planificación.

```
public void crawl() throws IOException{  
  
    URL url = new URL(this.urlToCrawl);  
  
    URLConnection urlConnection = null;  
    try {  
        urlConnection = url.openConnection();  
  
        try (InputStream input = urlConnection.getInputStream()) {  
  
            Document doc = Jsoup.parse(input, "UTF-8", "");  
            Elements elements = doc.select("a");  
  
            String baseUrl = url.toExternalForm();  
            for(Element element : elements){  
                String linkUrl = element.attr("href");  
                String normalizedUrl = UrlNormalizer.normalize(linkUrl, baseUrl);  
                crawler.linksQueue.put(normalizedUrl);  
  
                System.out.println(" - "+normalizedUrl);  
            }  
        }  
        if(crawler.barrier.getNumberWaiting()==1){  
            crawler.barrier.await();  
        }  
    }  
}
```

Figura 28 Función Crawler



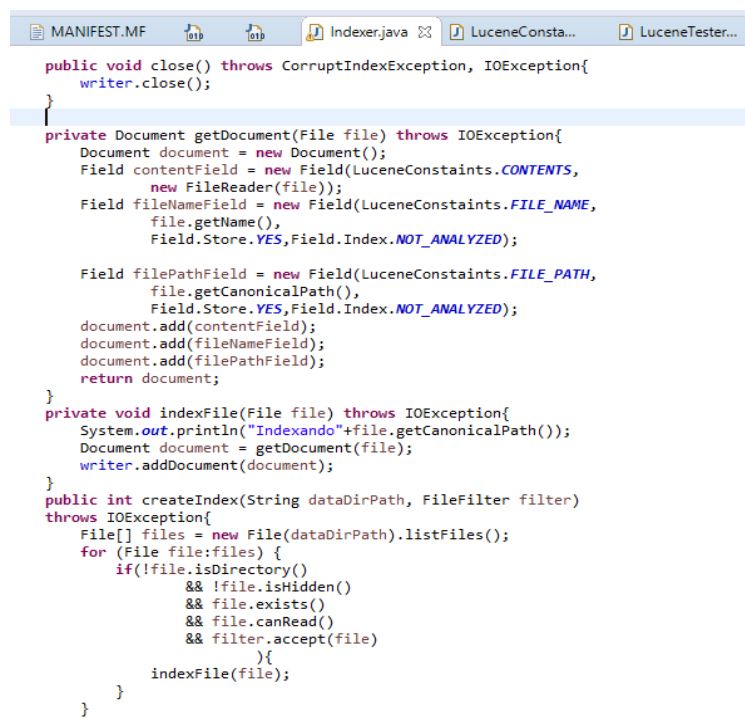
```

C:\Program Files\dotnet\dotnet.exe
Crawled j-balvin-bum-bum-tam-tam saved in Popular/reggaeton/j-balvin-bum-bum-tam-tam.txt
Crawled calle-13-latinoamerica saved in Popular/reggaeton/calle-13-latinoamerica.txt
Crawled chino-nacho-andas-en-mi-cabeza saved in Popular/reggaeton/chino-nacho-andas-en-mi-cabeza.txt
Crawled calle-13-ojos-color-sol saved in Popular/reggaeton/calle-13-ojos-color-sol.txt
Crawled carlos-rivera-que-lo-nuestro-se-queda-nuestro saved in Popular/reggaeton/carlos-rivera-que-lo-nuestro-se-queda-nuestro.txt
Crawled maluma-23 saved in Popular/reggaeton/maluma-23.txt
Crawled don-omar-208652 saved in Popular/reggaeton/don-omar-208652.txt
Crawled maluma-vente-pa-ca saved in Popular/reggaeton/maluma-vente-pa-ca.txt
Crawled carlos-rivera-fascinacion saved in Popular/reggaeton/carlos-rivera-fascinacion.txt
Crawled calle-13-respira-el-momento saved in Popular/reggaeton/calle-13-respira-el-momento.txt
Crawled j-balvin-sigo-extranandote saved in Popular/reggaeton/j-balvin-sigo-extranandote.txt
Crawled maluma-el-perdedor saved in Popular/reggaeton/maluma-el-perdedor.txt
Crawled j-balvin-si-tu-novio-te-deja-sola saved in Popular/reggaeton/j-balvin-si-tu-novio-te-deja-sola.txt
Crawled j-balvin-bonita saved in Popular/reggaeton/j-balvin-bonita.txt
Crawled daddy-yankee-havana saved in Popular/reggaeton/daddy-yankee-havana.txt
Crawled maluma-gps saved in Popular/reggaeton/maluma-gps.txt
Crawled calle-13-1357954 saved in Popular/reggaeton/calle-13-1357954.txt
Crawled j-balvin-ahora-dice saved in Popular/reggaeton/j-balvin-ahora-dice.txt
Crawled daddy-yankee-119501 saved in Popular/reggaeton/daddy-yankee-119501.txt
Crawled farruko-krippy-kush-remix-part-nicki-minaj-bad-bunny-21-savage-y-rvssian saved in Popular/reggaeton/farruko-krippy-kush-remix-part-nicki-minaj-bad-bunny-21-savage-y-rvssian.txt
Crawled maluma-trap saved in Popular/reggaeton/maluma-trap.txt
Crawled calle-13-611173 saved in Popular/reggaeton/calle-13-611173.txt
Crawled carlos-rivera-como-pagarte saved in Popular/reggaeton/carlos-rivera-como-pagarte.txt
Crawled carlos-rivera-gracias-a-ti saved in Popular/reggaeton/carlos-rivera-gracias-a-ti.txt
Crawled maluma-chantaje saved in Popular/reggaeton/maluma-chantaje.txt
Crawled farruko-quiereme saved in Popular/reggaeton/farruko-quiereme.txt
Crawled chino-nacho-1556793 saved in Popular/reggaeton/chino-nacho-1556793.txt

```

Figura 29 Búsqueda de URL's

El crawler que alimenta al proceso de extracción de texto, fue realizado en lenguaje de programación C# con visual studio 2017, la figura 28 muestra el método crawler el cual es el encargado de normalizar la URL al momento de abrir la conexión a la Web y dentro de la figura 29 se tiene el funcionamiento de la búsqueda de algunas URL's que visita el crawler.



```

MANIFEST.MF  Indexer.java  LuceneConsta...  LuceneTester...
public void close() throws CorruptIndexException, IOException{
    writer.close();
}

private Document getDocument(File file) throws IOException{
    Document document = new Document();
    Field contentField = new Field(LuceneConstaints.CONTENTES,
        new FileReader(file));
    Field fileNameField = new Field(LuceneConstaints.FILE_NAME,
        file.getName(),
        Field.Store.YES,Field.Index.NOT_ANALYZED);

    Field filePathField = new Field(LuceneConstaints.FILE_PATH,
        file.getCanonicalPath(),
        Field.Store.YES,Field.Index.NOT_ANALYZED);
    document.add(contentField);
    document.add(fileNameField);
    document.add(filePathField);
    return document;
}

private void indexFile(File file) throws IOException{
    System.out.println("Indexando"+file.getCanonicalPath());
    Document document = getDocument(file);
    writer.addDocument(document);
}

public int createIndex(String dataDirPath, FileFilter filter)
throws IOException{
    File[] files = new File(dataDirPath).listFiles();
    for (File file:files) {
        if(!file.isDirectory()
            && !file.isHidden()
            && file.exists()
            && file.canRead()
            && filter.accept(file)
        ){
            indexFile(file);
        }
    }
}

```

Figura 30 Función Indexar

Por otro lado, se tiene a la figura 30, que muestra el código fuente del método que sirve para la indexación de archivos por medio de la librería lucene, este método fue desarrollado en lenguaje java con ide eclipse el mismo que permite asociar una serie de archivos para posteriormente genera un índice. fdt con el contenido consolidado de los archivos indexados.

```
private void searchUsingFuzzyQuery(String searchQuery)
throws IOException, ParseException{
    searcher = new Searcher(indexDir);
    long startTime = System.currentTimeMillis();

    Term term = new Term(LuceneConstaints.CONTENTES, searchQuery);

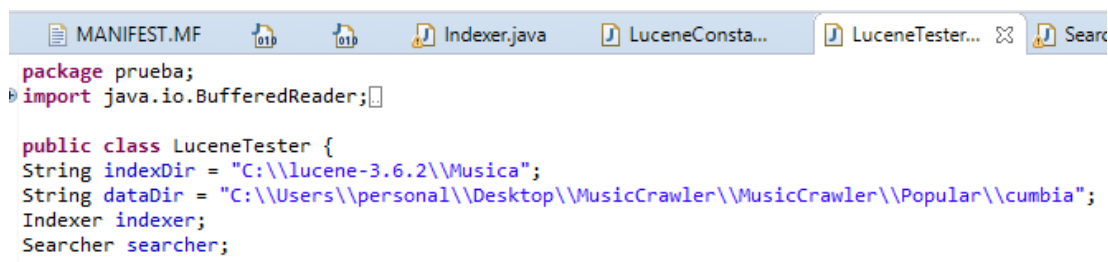
    Query query = new FuzzyQuery(term);

    TopDocs hits= searcher.search(query);
    long endTime= System.currentTimeMillis();

    System.out.println(hits.totalHits + "Documentos encontrados. Tiempo :"+ (endTime-startTime)+ "ms");
    for(ScoreDoc scoreDoc: hits.scoreDocs) {
        Document doc= searcher.getDocument(scoreDoc);
        System.out.println("Score:"+ scoreDoc.score + " ");
        System.out.println("File:"+ doc.get(LuceneConstaints.FILE_PATH));
    }
    searcher.close();
}
```

Figura 31 Función Buscar

Por otro lado, se muestra en la figura 31 el método buscar que incluye la librería lucene, el mismo que permite encontrar palabras específicas dentro del índice que contiene los documentos previamente indexados, es por esta razón que una vez que analiza el contenido de este archivo arroja dos características: i). la cantidad de archivos que contiene la palabra seleccionada, ii) el path con la ubicación del archivo encontrado.

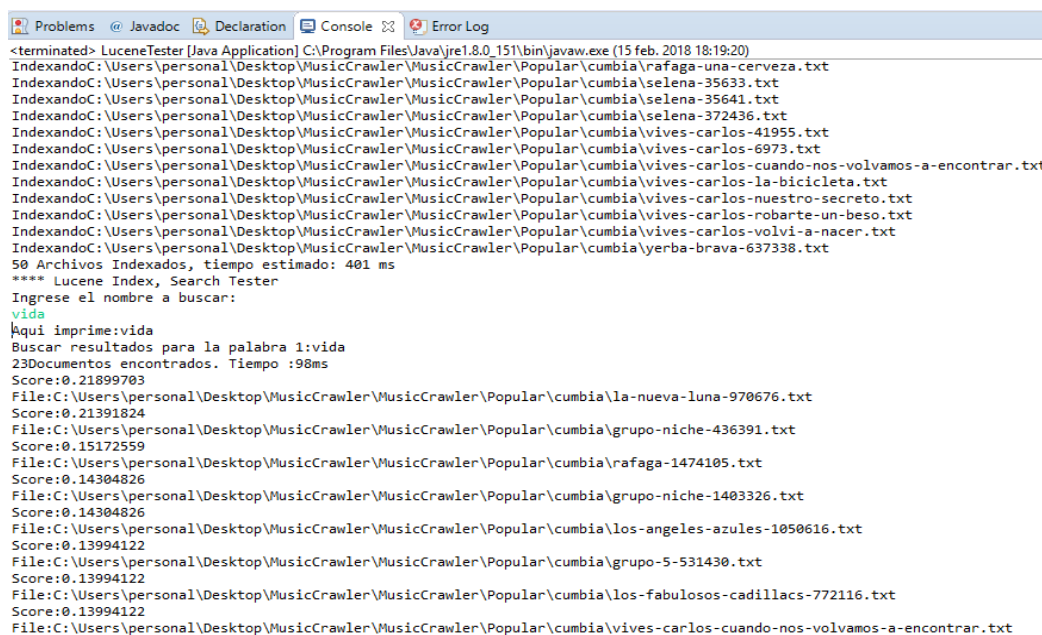


```
package prueba;
import java.io.BufferedReader;

public class LuceneTester {
    String indexDir = "C:\\\\lucene-3.6.2\\\\Musica";
    String dataDir = "C:\\\\Users\\\\personal\\\\Desktop\\\\MusicCrawler\\\\MusicCrawler\\\\Popular\\\\cumbia";
    Indexer indexer;
    Searcher searcher;
}
```

Figura 32 Función Indexar

Seguidamente, se justifica la codificación dentro de la figura 32, en la cual se observa como la función `indexar` que se maneja dentro de la librería `Lucene` permite la lectura de los archivos mediante el `path` de la carpeta en la que se encuentran los documentos con formato `.txt`, los mismos que fueron creados con anterioridad mediante el `crawler`. Por otro lado, el método de test incluye de igual manera el `path` que contiene la salida de los archivos indexados para su posterior análisis.



```

<terminated> LuceneTester [Java Application] C:\Program Files\Java\jre1.8.0_151\bin\javaw.exe (15 feb. 2018 18:19:20)
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\rafaga-una-cerveza.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\selena-35633.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\selena-35641.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\selena-372436.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-41955.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-6973.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-cuando-nos-volvamos-a-encontrar.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-la-bicicleta.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-nuestro-secreto.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-robarte-un-beso.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-volvi-a-nacer.txt
IndexandoC:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\yerba-brava-637338.txt
50 Archivos Indexados, tiempo estimado: 401 ms
**** Lucene Index, Search Tester
Ingrese el nombre a buscar:
vida
Aquí imprime:vida
Buscar resultados para la palabra 1:vida
23 Documentos encontrados. Tiempo :98ms
Score:0.21899703
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\la-nueva-luna-970676.txt
Score:0.21391824
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\grupo-niche-436391.txt
Score:0.15172559
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\rafaga-1474105.txt
Score:0.14304826
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\grupo-niche-1403326.txt
Score:0.14304826
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\los-angeles-azules-1050616.txt
Score:0.13994122
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\grupo-5-531430.txt
Score:0.13994122
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\los-fabulosos-cadillacs-772116.txt
Score:0.13994122
File:C:\Users\personal\Desktop\MusicCrawler\MusicCrawler\Popular\cumbia\vives-carlos-cuando-nos-volvamos-a-encontrar.txt

```

Figura 33 Resultados Lucene

Las clases `indexar` y `buscar` arrojan los resultados que contiene la figura 33, como se mencionó con anterioridad la primera parte consta de la indexación de archivos y el arrojamiento del número de documentos indexados seleccionados por la lectura del `path` de la carpeta en la que se encuentran y la segunda parte consta de la búsqueda de palabras repetidas en el mayor número de documentos indexados. Para ambos casos se muestra el tiempo que tomó realizar cada tarea.

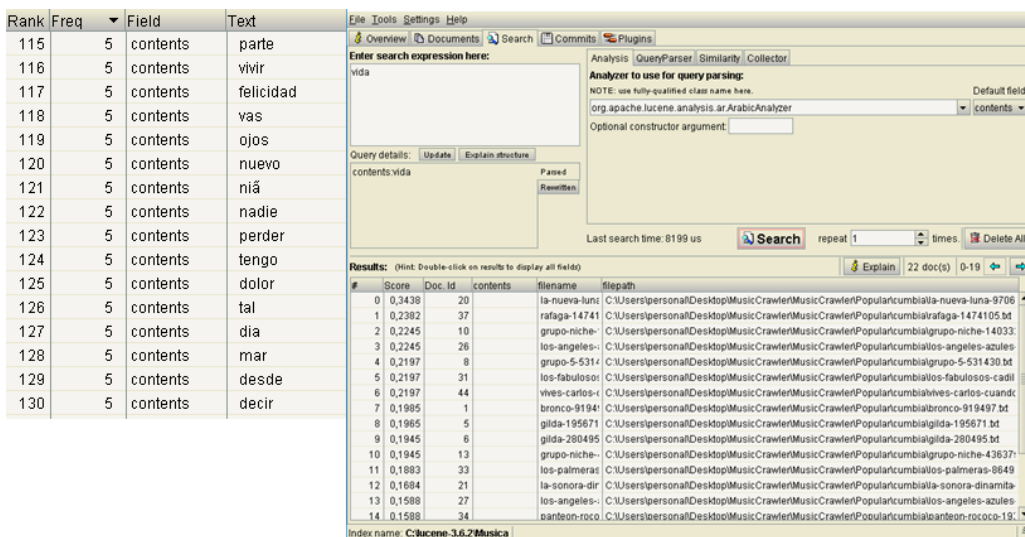


Figura 34 Frecuencia de datos

Para continuar con el proceso de análisis de texto se tiene las palabras con mayor frecuencia tal como se muestra en la figura 34, una vez que se extrae los términos se crea el vector idf para cada género musical, con los datos obtenidos se genera el archivo .csv para posteriormente analizar el texto con el método de clustering dentro de Weka, cabe mencionar que las pruebas que se realiza con cada algoritmo forman parte del siguiente apartado de la investigación.

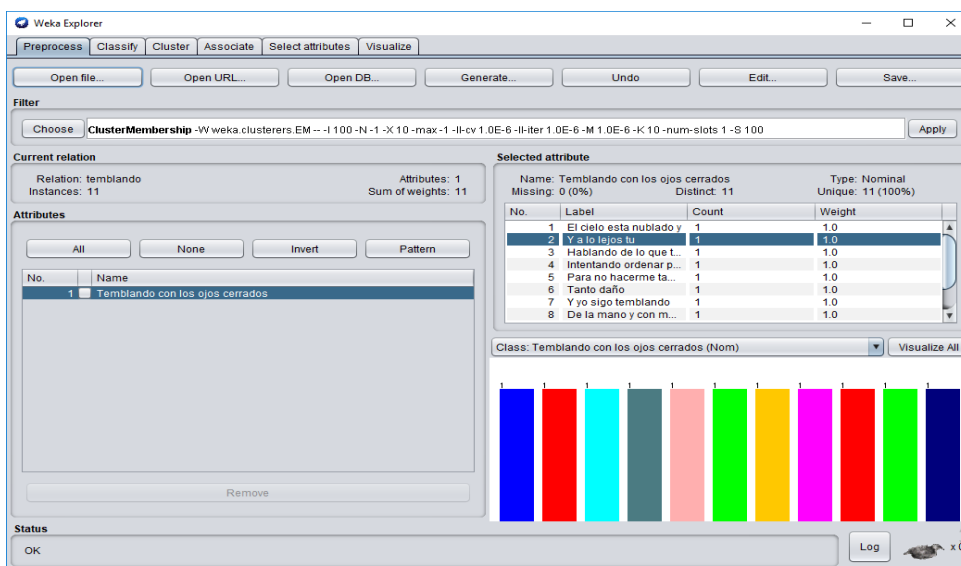


Figura 35 Análisis de texto-Preprocesamiento Weka

```

Clusterer output

kMeans
=====

Number of iterations: 2
Within cluster sum of squared errors: 9.0

Initial starting points (random):

Cluster 0: 'Tanto daño'
Cluster 1: 'El cielo esta nublado y'

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute                Full Data                Cluster#
                        (11.0)                  (10.0)
-----
Temblando con los ojos cerrados  El cielo esta nublado y  Y a lo lejos tu El cielo es

Time taken to build model (full training data) : 0 seconds

=== Evaluation on test set ===
Clustered Instances

```

Figura 36 Clúster Algoritmo K-means

Reanudando con el enunciado respecto al desarrollo del software, se tiene como siguiente instancia el proceso de análisis de texto que comienza por el preprocesamiento de datos como se presenta en la figura 35, seguidamente se tiene a la figura 36 que muestra el estudio de datos mediante el aprendizaje supervisado con el algoritmo simple k-means, el mismo que con la ayuda de la herramienta Weka permitió colocar una correcta ponderación a cada género musical mediante el estudio de las letras de sus canciones, lo que permitió obtener como salida los datos que consumirá el sistema propuesto en la investigación.

3.5. Conclusiones

Una de las características que diferencian a un crawler distribuido de un centralizado es la percepción de búsqueda, mientras el distribuido maneja una gran cantidad de URL's muchas de ellas sin estar relacionadas con la raíz, el centralizado realiza una búsqueda específica en base a la url raíz.

Dentro de la codificación que posee la herramienta Weka, se manifiesta que únicamente el método clúster con el algoritmo SimpleKmeans se basa en un número específico de clústers iniciales, para realizar el proceso de análisis de un conjunto de datos.

La biblioteca lucene permite una correcta indexación debido a las características que contiene cada clase que la componen, la solución más optima es el manejo de un índice que no es más que una estructura de datos que facilita la administración de la información.

CAPITULO 4

PRUEBAS Y ANÁLISIS DE RESULTADOS

Dentro de este capítulo se pretende llegar a la recolección de resultados de la investigación, como el tema que se ha venido estudiando consta de algunos procesos, se ha realizado una observación de cada uno de ellos por separado, dentro del proceso de extracción de datos no se considera necesario especular los datos obtenidos, se continúa con las pruebas dentro del proceso de análisis de datos.

Para el estudio de datos previamente extraídos surgen algunas características que se valora para la toma de decisiones sobre la selección del mejor algoritmo dentro de la herramienta Weka, es por esta razón que dicho procedimiento requiere de una discusión de los resultados debido a que de la elección del mejor algoritmo se procederá a continuar con el siguiente proceso, para que cada una de las técnicas desarrolladas funcione satisfactoriamente se requiere de una recopilación de conceptos técnicos que ayudan al correcto funcionamiento y manipulación de cada una sustentando las valoraciones que arrojan con un determinado conjunto de datos.

Para tener una evaluación completa sobre el aplicativo web, se presenta dentro de este apartado las pruebas funcionales que muestra el último proceso, siendo este la recolección de todas las técnicas previamente analizadas, seleccionadas y justificadas, se simula dentro de cada módulo el correcto flujo de acciones que el usuario deberá manejar para que el proceso funcione correctamente, así como un flujo de acciones alternas que tanto el usuario como la aplicación podrán arrojar al momento de manipular cada módulo descrito por SCRUM en el apartado anterior.

El último proceso que constituye en la manipulación del aplicativo web no presenta la necesidad de una discusión sobre los datos que lo componen, más bien estará sujeta a discusión en cuanto a la usabilidad y aceptación que tendrá en el campo en el que será puesto a producción, es por esta razón que se muestra las tabulaciones de la encuesta descriptiva sobre una población selecta.

4.1. Pruebas de Análisis de Datos

Las pruebas del estudio de datos se manejaron con la herramienta Weka, la misma que contiene diferentes entornos como se mencionó con anterioridad al momento de justificar la selección de la misma, con dichos antecedentes, se seleccionó el entorno visual EXPLORER ya que se presta a una manipulación de datos mucho más accesible al momento de accionar los algoritmos que contiene el método *clúster*.

El siguiente estudio es el resultado del análisis de cada algoritmo que contiene el método clúster de Weka, para una correcta toma de decisiones, en cuando al mejor algoritmo se basó en el porcentaje de instancias incorrectamente agrupadas, es por esta razón que mientras menor sea el porcentaje más eficiente será el algoritmo, y en base a esta selección se procedió a escoger el grupo de datos que se encuentren dentro de los clúster que arrojó el mejor algoritmo, a continuación se muestra el conjunto de clúster enlazados con cada dato arrojado así como el porcentaje de instancias y el diagrama de clusterización de cada uno.

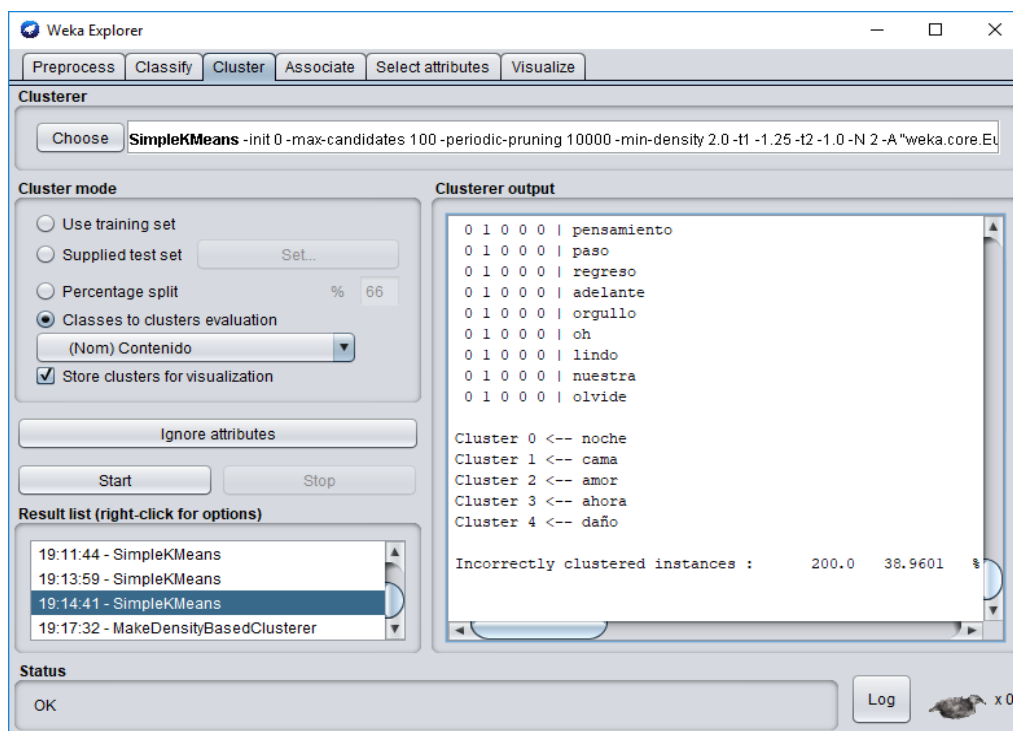


Figura 37 Simple K-means

La figura 38 contiene el resultado del método clúster con el algoritmo SimpleKmeans, el mismo que fue accionado con un numero de clúster 5 para el análisis de texto, obteniendo dentro del listado las cinco primeras palabras localizadas en cada clúster: noche, cama, amor, ahora y daño. El algoritmo tuvo un 38.9601% de instancias incorrectas de todo el conjunto de datos examinados.

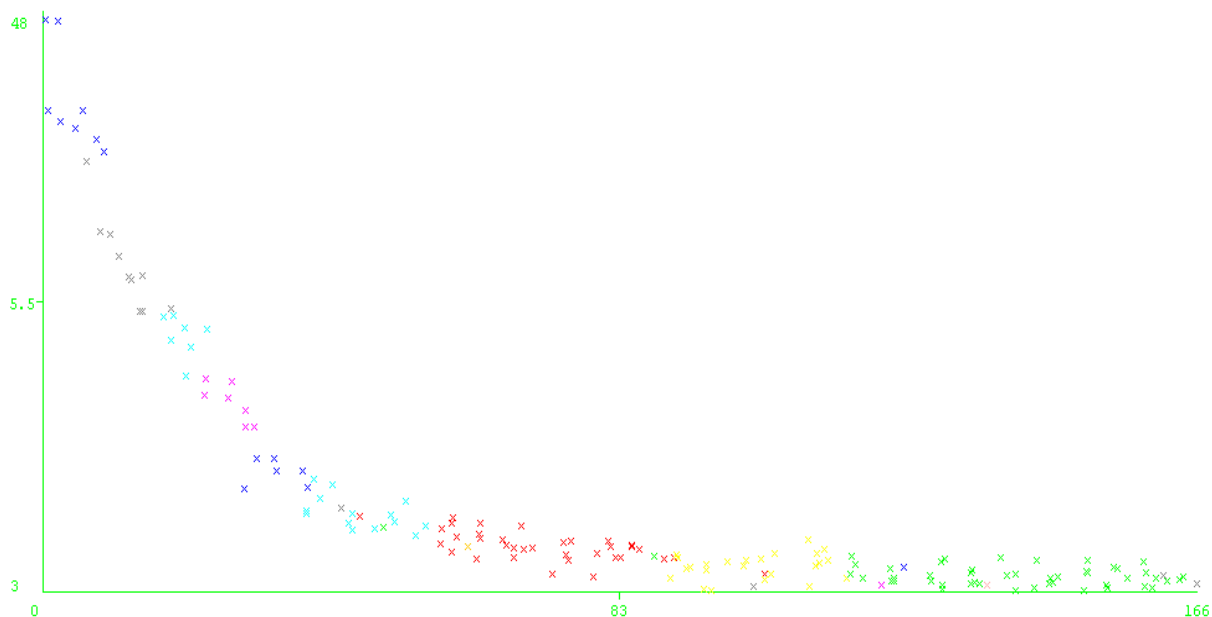


Figura 38 Diagrama Algoritmo SimpleKmean

En la figura 39 se muestra el diagrama que arroja el algoritmo al momento de generar los clústers, este proceso se obtiene de forma automática, puesto que es el resultado del progreso que tuvo cada iteración, este gráfico es un complemento que posee internamente la herramienta Weka.

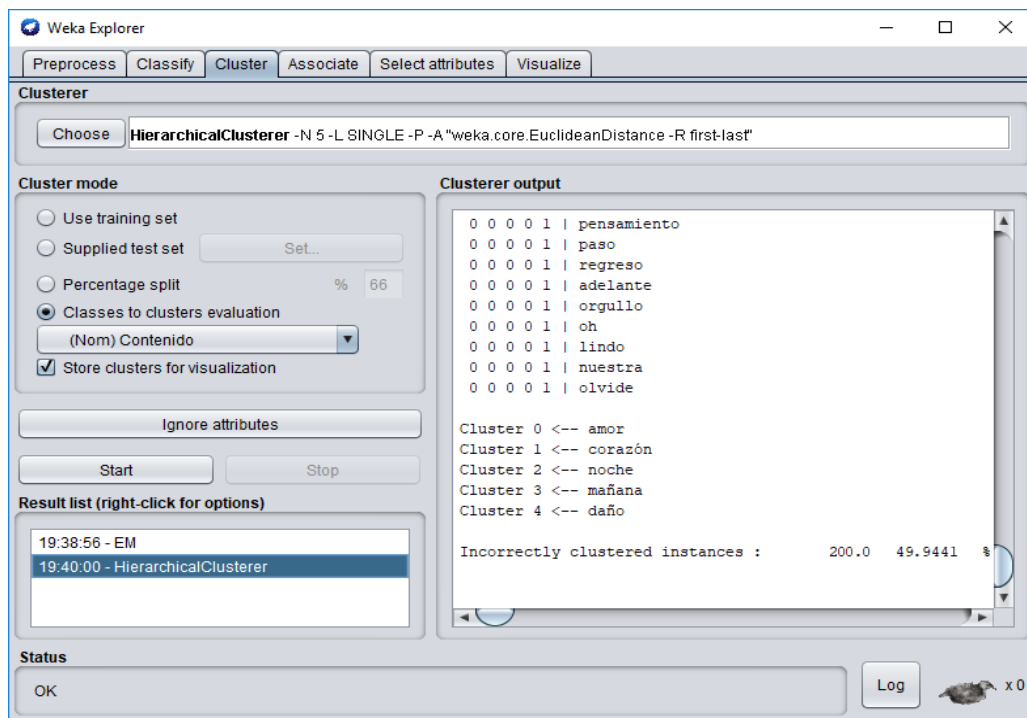


Figura 39 Algoritmo HierarchicalClusterer

La figura 40 contiene el resultado del método clúster con el algoritmo HierarchicalClusterer, el mismo que fue accionado con un numero de clúster 5 para el análisis de texto, obteniendo dentro del listado las cinco primeras palabras localizadas en cada clúster: amor, corazón, noche, mañana y daño. El algoritmo tuvo un 49.9441% de instancias incorrectas de todo el conjunto de datos examinados.

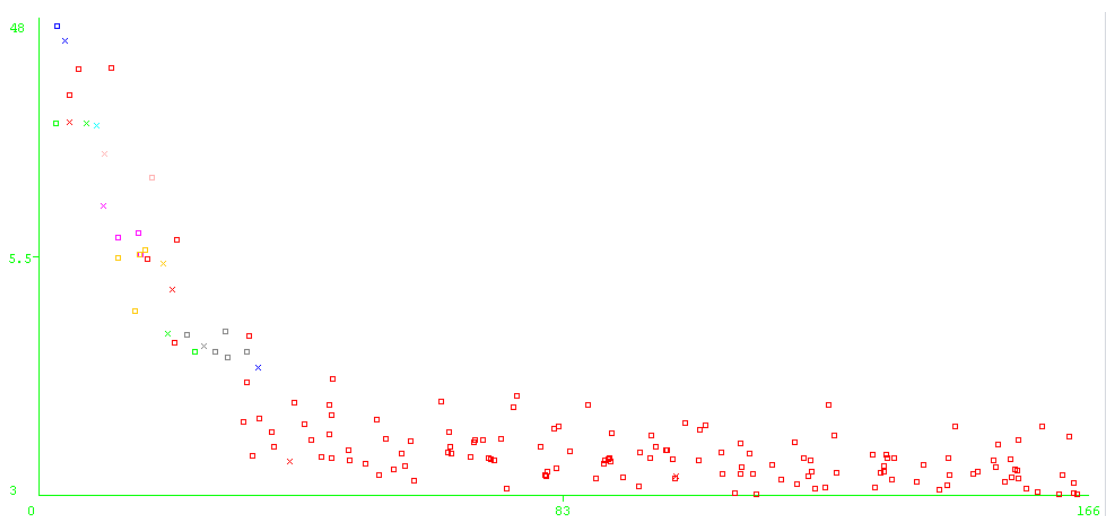


Figura 40 Diagrama Algoritmo HierarchicalClusterer

En la figura 41 se muestra el diagrama que arroja el algoritmo al momento de generar los clústers, este proceso se obtiene de forma automática, puesto que es el resultado del progreso que tuvo cada iteración, este gráfico es un complemento que posee internamente la herramienta Weka.

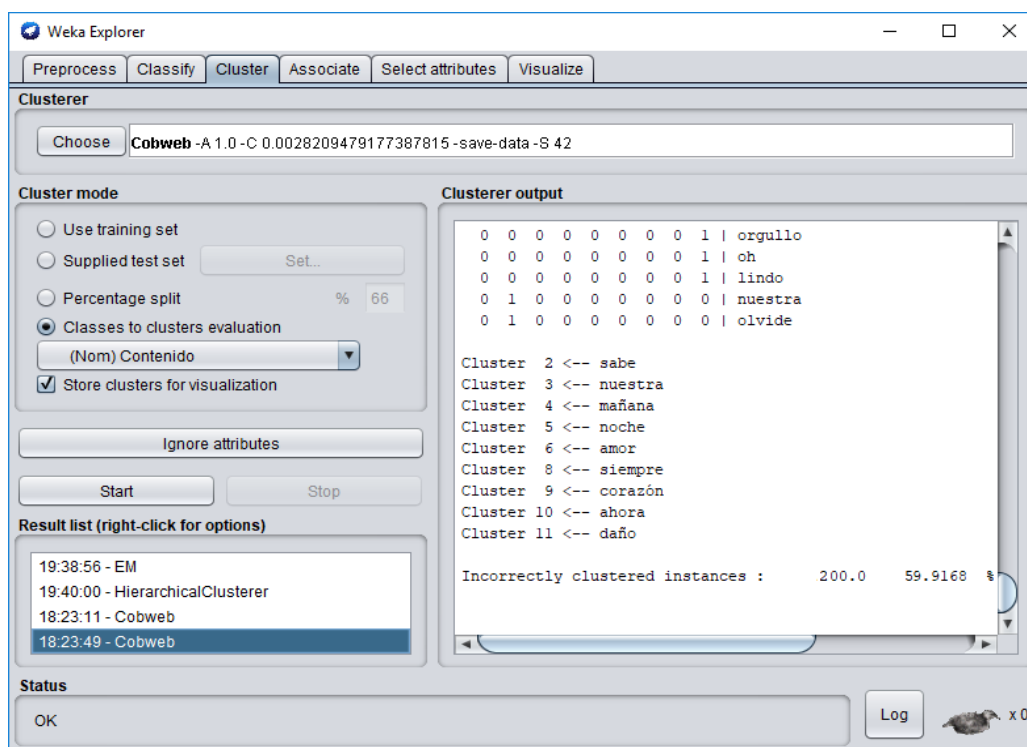


Figura 41 Algoritmo Cobweb

La figura 42 contiene el resultado del método clúster con el algoritmo Cobweb, el mismo que fue accionado con un numero de clúster 9 para el análisis de texto, obteniendo dentro del listado las cinco primeras palabras localizadas en cada clúster: sabe, nuestra, mañana, noche y amor. El algoritmo tuvo un 59.9168% de instancias incorrectas de todo el conjunto de datos examinados.

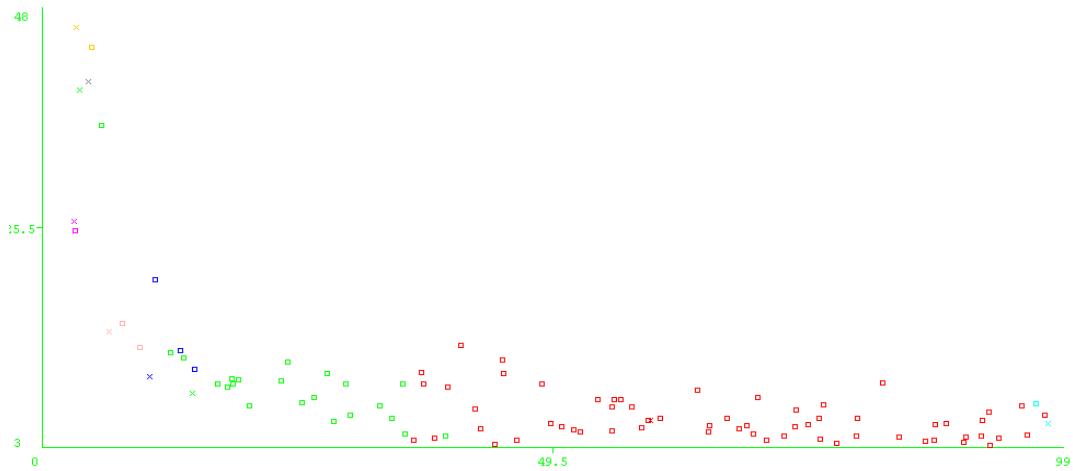


Figura 42 Diagrama Algoritmo Cobweb

En la figura 43 se muestra el diagrama que arroja el algoritmo al momento de generar los clústers, este proceso se obtiene de forma automática, puesto que es el resultado del progreso que tuvo cada iteración, este gráfico es un complemento que posee internamente la herramienta Weka. Para ver si guarda

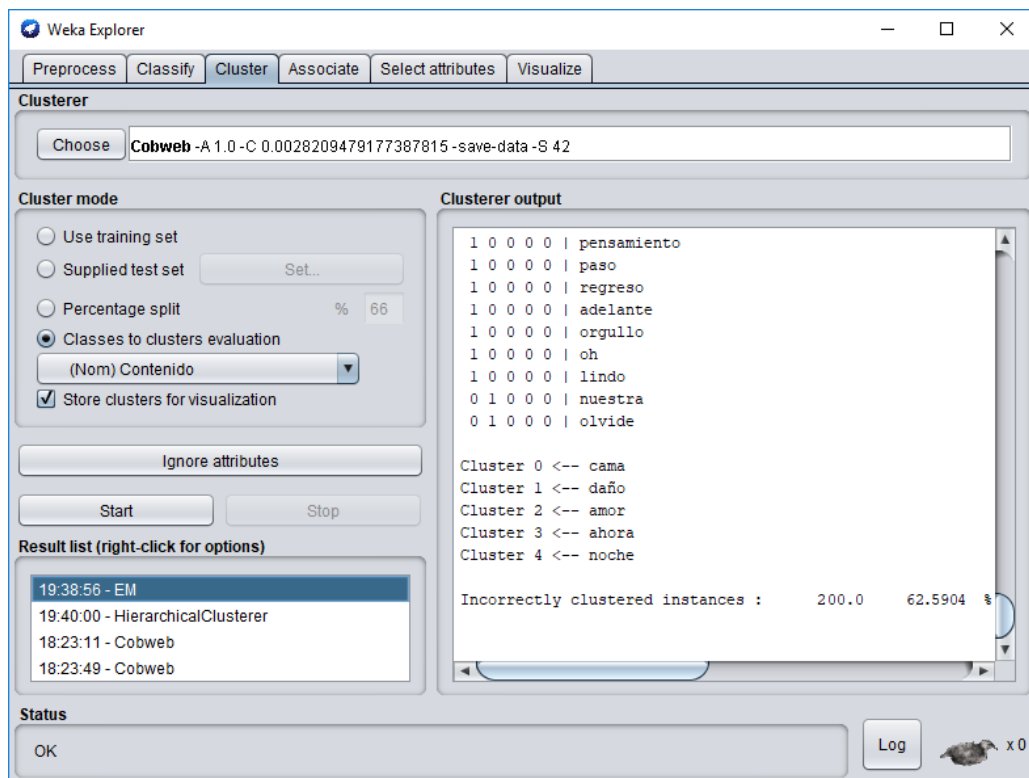


Figura 43 Algoritmo EM

La figura 44 contiene el resultado del método clúster con el algoritmo Cobweb, el mismo que fue accionado con un numero de clúster 5 para el análisis de texto, obteniendo dentro del listado las cinco primeras palabras localizadas en cada clúster: cama, daño, amor, ahora y noche. El algoritmo tuvo un 62.5904% de instancias incorrectas de todo el conjunto de datos examinados.

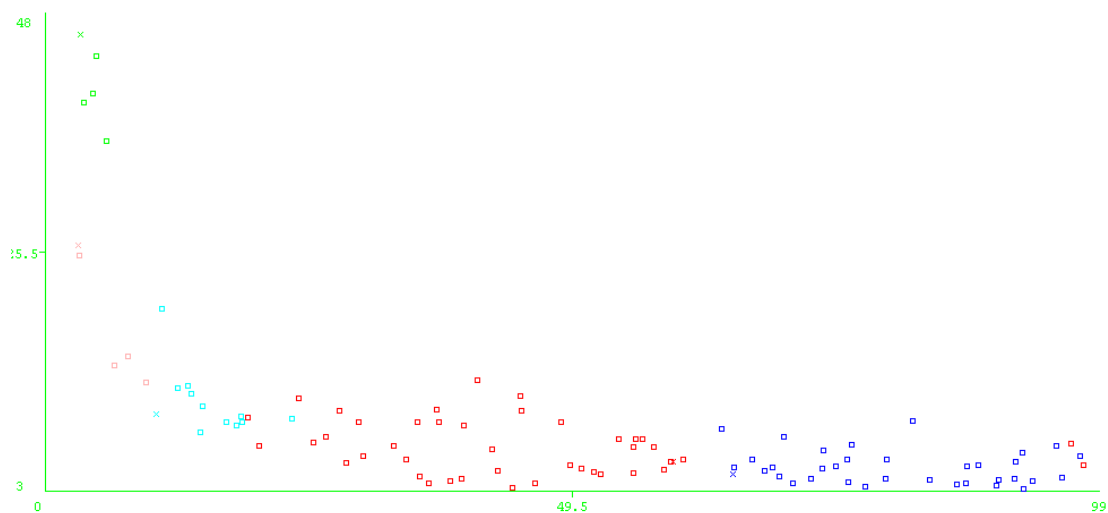


Figura 44 Diagrama Algoritmo EM

En la figura 45 se muestra el diagrama que arroja el algoritmo al momento de generar los clústers, este proceso se obtiene de forma automática, puesto que es el resultado del progreso que tuvo cada iteración, este gráfico es un complemento que posee internamente la herramienta Weka.

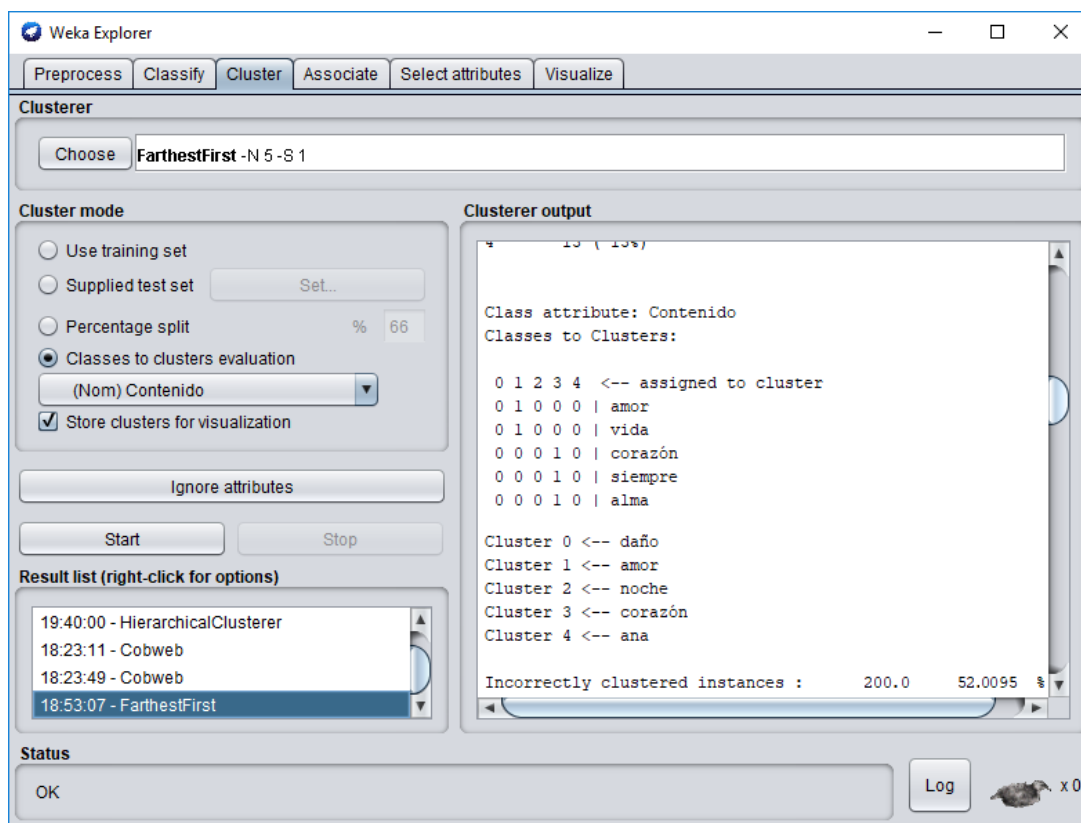


Figura 45 Algoritmo FarthestFirst

La figura 46 contiene el resultado del método clúster con el algoritmo Cobweb, el mismo que fue accionado con un numero de clúster 5 para el análisis de texto, obteniendo dentro del listado las cinco primeras palabras localizadas en cada clúster: daño, amor, noche, corazón y ana. El algoritmo tuvo un 52.0095% de instancias incorrectas de todo el conjunto de datos examinados.

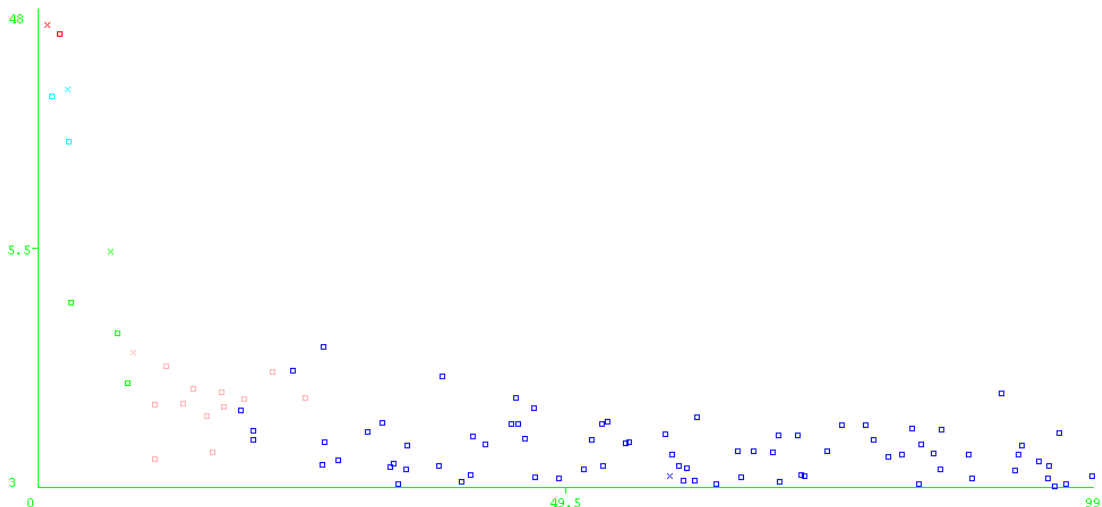


Figura 46 Diagrama Algoritmo FarthestFirst

En la figura 47 se muestra el diagrama que arroja el algoritmo al momento de generar los clústers, este proceso se obtiene de forma automática, puesto que es el resultado del progreso que tuvo cada iteración, este gráfico es un complemento que posee internamente la herramienta Weka.

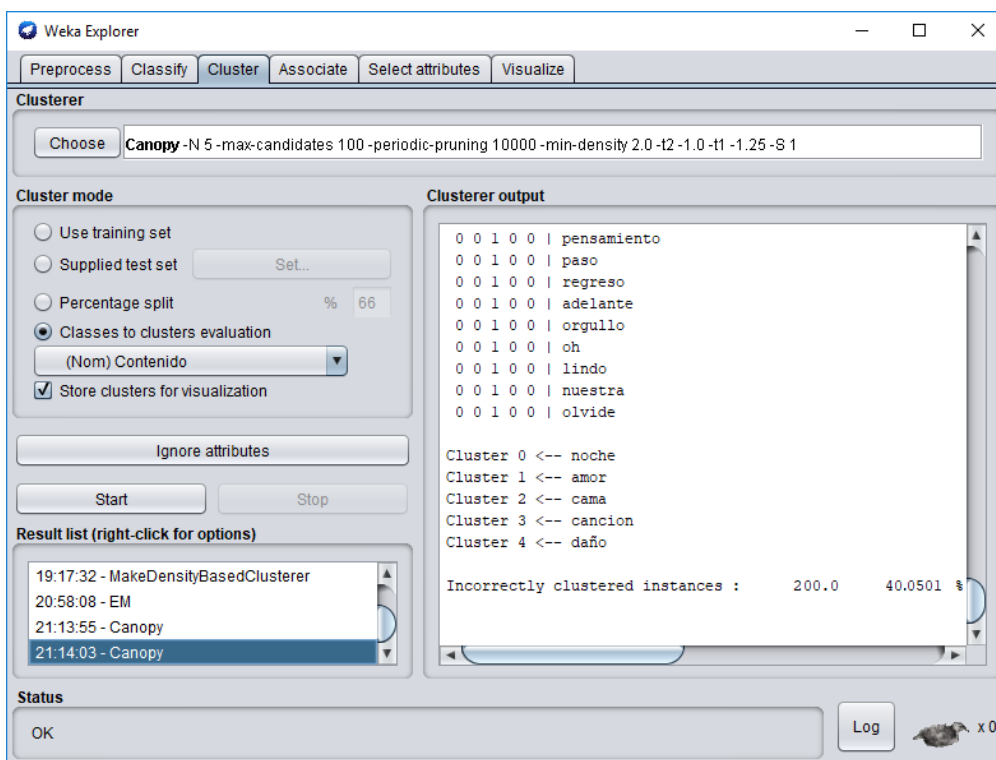


Figura 47 Algoritmo Canopy

La figura 48 contiene el resultado del método clúster con el algoritmo Canopy, el mismo que fue accionado con un numero de clúster 5 para el análisis de texto, obteniendo dentro del listado las cinco primeras palabras localizadas en cada clúster: noche, amor, cama, corazón y daño. El algoritmo tuvo un 40.0501% de instancias incorrectas de todo el conjunto de datos examinados.

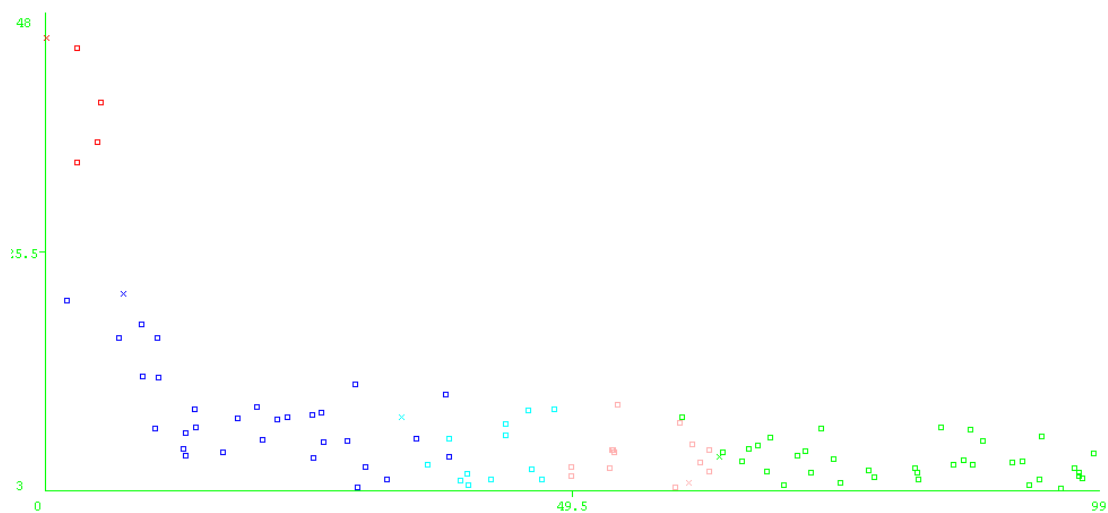


Figura 48 Diagrama del Algoritmo Canopy

En la figura 49 se muestra el diagrama que arroja el algoritmo al momento de generar los clústers, este proceso se obtiene de forma automática, puesto que es el resultado del progreso que tuvo cada iteración, este gráfico es un complemento que posee internamente la herramienta Weka.

4.1.1. Discusión de resultados

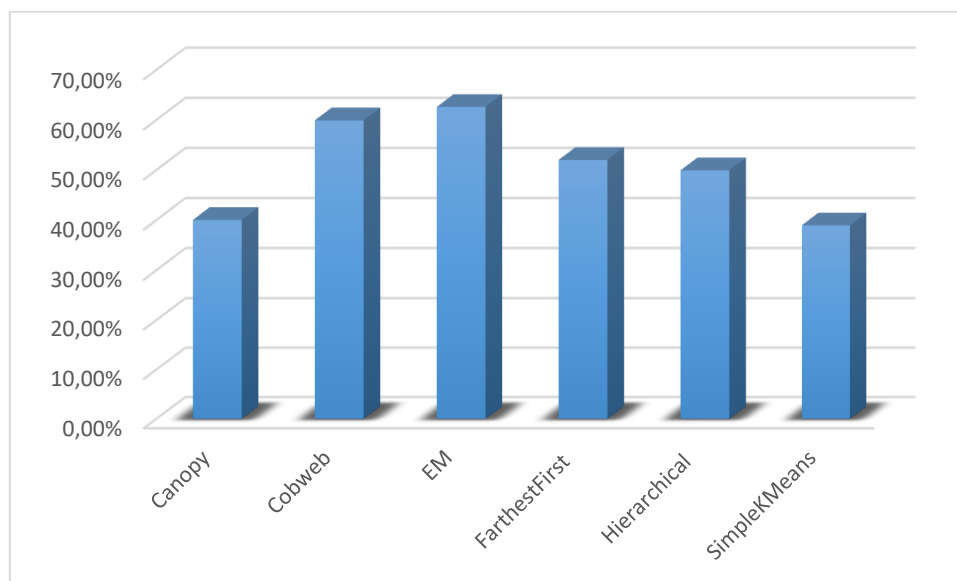


Figura 49 Algoritmos Método Clúster Weka

Las pruebas que se realizó con cada algoritmo dentro del método clúster de Weka arrojó diferentes valores para una discusión sobre el mejor algoritmo que podría presentar datos con menos error, en la figura 50 se muestra los porcentajes de acuerdo a cada uno de los algoritmos y se justifica la selección del método clúster con el algoritmo SimpleKmean ya que este fue el que obtuvo un menor porcentaje de instancias incorrectas dentro de todo el conjunto de datos analizados, a raíz de esta selección se extrae el contenido para la base de datos del sistema mediante la opción de Weka al usar el conjunto de entrenamiento para los datos seleccionados.

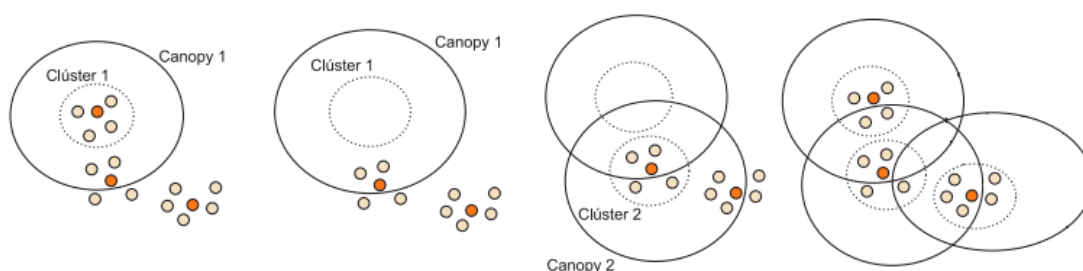


Figura 50 Funcionamiento del Algoritmo Canopy

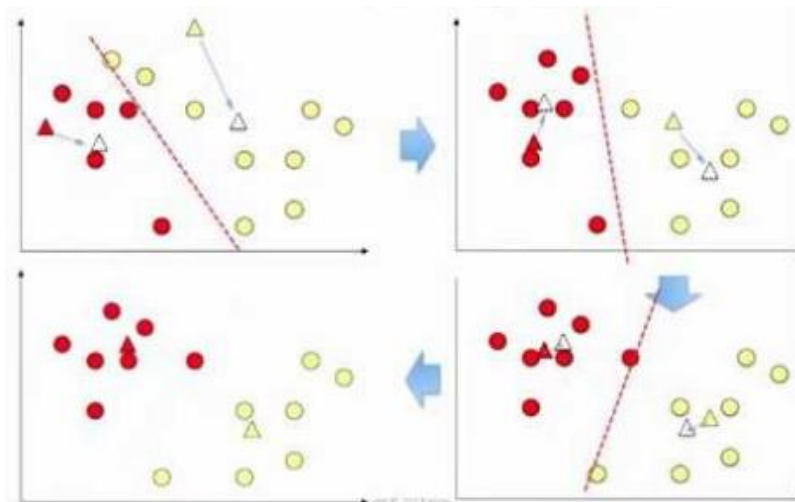


Figura 51 Funcionamiento del Algoritmo Simple Kmeans

Por otro lado, se observa que el algoritmo Canopy de clustering jerárquico y SimpleKmeans de clustering particional son los que menor porcentaje de error arrojan, y dentro del análisis se puede observar la diferente forma de clusterizar de cada uno, mientras Canopy trata de determinar los centroides aproximados del grupo usando dos valores de distancia, $T1$ y $T2$, con $T1 > T2$., no devuelve valores exactos, creando grupos de forma muy rápida (ver figura 51) pudiendo dar el número óptimo de los grupos sin siquiera especificar el número k como se requiere en k -means, por otro lado k -means al necesitar que se establezca un número k de grupos como parámetro de entrada realiza dos pasos en base al número señalado: el primero es encontrar los puntos más cercanos a cada centroide y los asigna a un grupo específico; el segundo paso vuelve a calcular el punto centroide utilizando el promedio de las coordenadas de todos los puntos de ese clúster (ver figura 52) (Ormeño Silva, 2014).

4.2. Pruebas funcionales del sistema

Dentro de todo proceso de desarrollo del software la tarea más compleja no siempre es la codificación del producto como tal, sino que la tarea más dura termina siendo el proceso de pruebas por el que debe pasar el producto para poder entrar en producción cumpliendo los lineamientos planteados dentro del proceso de análisis de requerimientos para satisfacer las necesidades que intenta resolver el correcto funcionamiento del producto software. Las pruebas de un producto software no son

más que una verificación del comportamiento del sistema en base a un conjunto de ejecuciones seleccionadas o a su vez la ejecución de casos de pruebas.

En este apartado se pretende exponer las pruebas que se realizó a los procesos de recolección y análisis de datos previo a que estos puedan ser consumidos en la aplicación como tal, es por esta razón, que a continuación se muestran los casos de prueba funcional de los módulos: Web Crawler, Weka, Aplicación Web.

Tabla20

Caso de prueba módulo web crawler

Código de Identificación	Sistem.exam. web01
Nombre del caso de prueba	Funcionalidad de Web Crawler
Descripción	El administrador podrá comprobar la descarga de los archivos visitando las carpetas por género musical de los archivos con formato .txt almacenados localmente en el computador. C:\Users\MusicCrawler\Popular
Variables de entrada	Dirección de la página web a visitar
Flujo Normal del Evento	<ol style="list-style-type: none"> 1. Lectura de la dirección de la página web. 2. Examina el contenido de la página web con sus diferentes links que se encuentran enlazados. 3. Extrae la letra de las canciones y las almacena en el directorio local del computador.
Resultado esperado	Se puede manipular los archivos txt dentro de cada carpeta del género musical.
Flujo Alterno	<ol style="list-style-type: none"> 1. El link encontrado no contiene ninguna cabecera de la página web con contenido relacionado a letras de canciones.
Resultado alternativo esperado	El sistema da por terminado con esa raíz y continua con la siguiente.

Tabla21

Caso de prueba módulo weka

Código de Identificación	Sistem.exam. web02
---------------------------------	---------------------------

CONTINÚA



Nombre del caso de prueba	Funcionalidad de Lucene-Weka
Descripción	El programa recupera los documentos .txt para indexarlos en un solo tipo de archivo. fdt para poder procesarlos y extraer el contenido con las palabras de mayor frecuencia de esta manera se puede generar el documento .cvs para procesarlo en Weka.
Variables de entrada	Archivos con formato txt clasificados en una carpeta por género musical
Flujo Normal del Evento	<ol style="list-style-type: none"> 1. Lectura de archivos ubicados localmente dentro de una carpeta del computador C:\ \Popular\reggaetón. 2. Indexación de archivos seleccionados y almacenar en un archivo con formato. fdt. 3. Agrupar el contenido por la frecuencia de repetición de cada palabra en el archivo. fdt. 4. Seleccionar las 200 primeras palabras para generar el archivo .cvs. 5. Procesar el archivo .cvs con el algoritmo SimpleKmeans de agrupamiento que posee Weka
Resultado esperado	Directorio de archivos. arff para almacenar en la base de datos del sistema.
Flujo Alternativo	Indexar los archivos ubicados localmente que contienen el formato .txt de cada letra de canción agrupada por género musical.
Resultado alternativo esperado	Almacenamiento local de los archivos con formato. fdt.

Tabla22

Caso de prueba módulo aplicación web

Código de Identificación	Sistem.exam. web03
Nombre del caso de prueba	Funcionalidad de Aplicativo Web
Descripción	La aplicación muestra en cada pestaña de la página web el contenido de los datos previamente analizados.
Variables de entrada	Archivos con formato SQL clasificados en una base de datos separadas por popularidad y en cada una de ellas clasificadas por género musical
Flujo Normal del Evento	<ol style="list-style-type: none"> 1. Lectura de contenido registrado en la base de datos. 2. Visualización de datos en la página web.

CONTINÚA



Resultado esperado

Manipulación de datos por el usuario final que muestra la página web.

4.3. Tabulación de la Encuesta

Uno de los objetivos de la investigación que finaliza es evaluar de manera estadística el grado de satisfacción que obtuvo el sistema desarrollado una vez que ha finalizado el proceso de pruebas, y entra a una etapa de simulación de la puesta en marcha del producto. La información fue recogida mediante encuestas en línea una vez que los estudiantes finalizaron con la manipulación de la página web desarrollada, cabe mencionar que se ha obtenido una población de 65 estudiantes de los alumnos de la Carrera de Ingeniería en Sistemas incluyendo dentro de la misma a un total de 5 músicos que aportaron con esta fase.

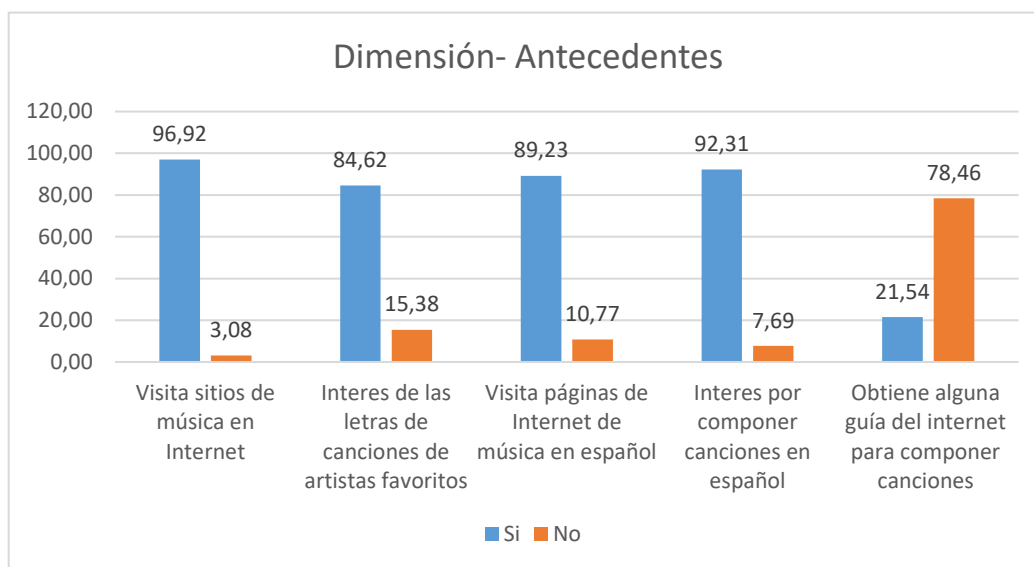


Figura 52 Valoración Dimensión Antecedentes

Como se observa en la figura 53, varios estudiantes visitan sitios de música en Internet, de igual manera un 84.62% muestran interés por las letras de sus artistas favoritos, cabe recalcar de más de un 80% de las personas encuestadas afirman que visitan sitios web de música en español, por otro lado, un 92.31% muestran interés por llegar a componer canciones en español, pero a su vez confirman que no obtienen ninguna guía dentro de los sitios web que visitan con regularidad.

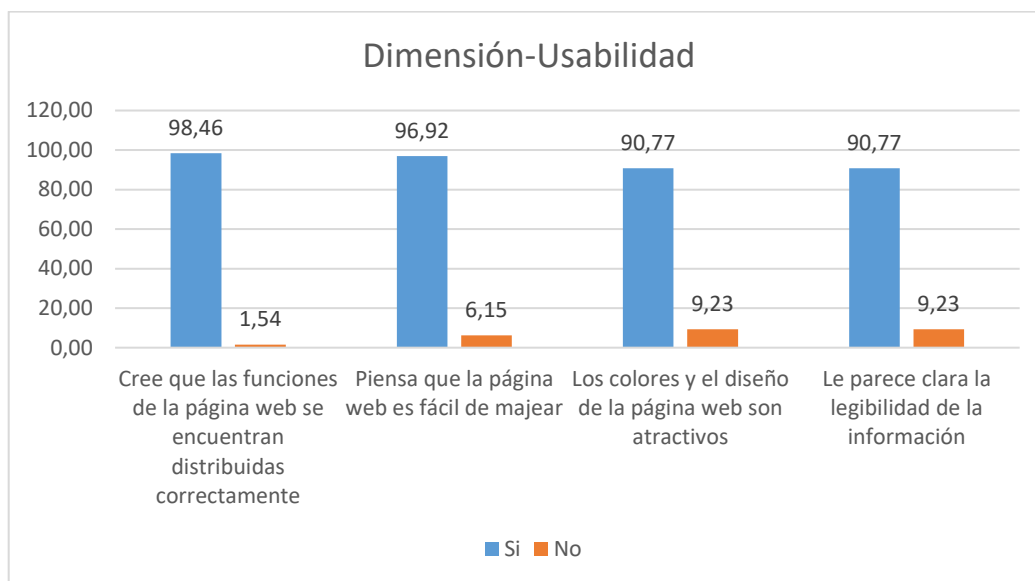


Figura 53 Valoración Dimensión Usabilidad

Dentro del análisis que arroja la encuesta sobre la dimensión de usabilidad podemos ver (figura 54) que la mayoría de la población acoge positivamente a la página web, este resultado se basa en que los factores de usabilidad como la eficacia que contiene tanto la distribución y el manejo del contenido de la página web, así como el patrón de satisfacción que contiene el diseño y la legibilidad de la información se encuentran en una valoración superior al 90%, es por esta razón que el cometido de la página web en cuanto a los factores de usabilidad, ha sido evaluado como satisfactorio para la etapa de simulación de puesta en marcha de la aplicación.

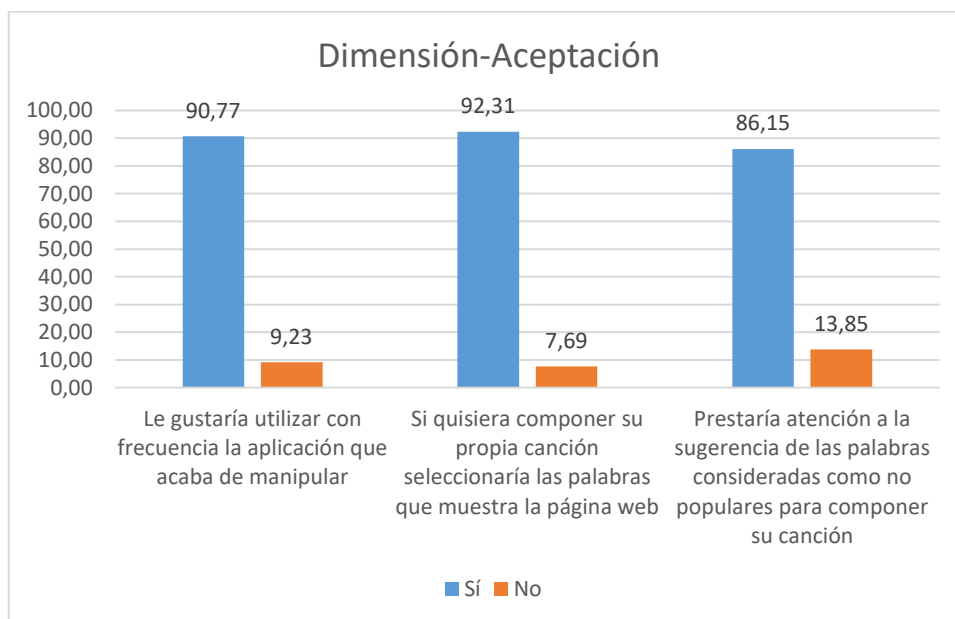


Figura 54 Valoración Dimensión Aceptación

Para tener una estimación en valores sobre la aceptación que podría tener la página web ciertas preguntas dentro de la encuesta abarcaron sobre esta dimensión (ver figura 55), arrojando un 90.77% sobre el uso con frecuencia de la aplicación, así como un 92.31% de la población aceptaría las recomendación sobre el texto que muestra la aplicación para componer su propia canción en español, pero se encontró que un 86.15% de la población no aceptaría el texto recomendado sobre las canciones de los géneros menos populares, esto conlleva a dejar una inquietud sobre la popularidad de las canciones y los clásicos que cada género musical contiene.

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

Para culminar con la investigación que abarca los temas de la extracción de texto y el análisis de la información en base a los métodos del procesamiento de lenguaje natural se presenta a continuación una serie de conclusiones y recomendaciones, así como una línea de trabajos a futuro.

5.1. Conclusiones

La selección adecuada de recursos provenientes de la Web permitió acercarse a una construcción óptima del web crawler, facilitando la elección del tipo de búsqueda que sirvió como referencia para la programación del almacenamiento de texto.

La preferencia por un método de análisis de texto óptimo y la manipulación pertinente de algoritmos no supervisados dentro de un conjunto de datos, permite dar mayor confiabilidad al usuario en el momento de operar el prototipo desarrollado.

Los patrones de usabilidad fueron diseñados acatando las buenas normas de desarrollo; el diseño, la distribución y la legibilidad de la información fueron tabuladas dentro de las encuestas descriptivas arrojando una aprobación de alrededor del 90% de la población encuestada.

Dentro del proceso de pruebas más del 80% de la población encuestada acoge el prototipo de manera positiva, dando como resultado que las recomendaciones musicales clasificadas por género son aceptadas por los usuarios finales.

El uso de la librería lucene dentro del proceso de desarrollo permitió la indexación de archivos arrojados por el web crawler logrando la agrupación de datos de acuerdo a la frecuencia de repetición, lo que facilitó el manejo de información para continuar con el proceso de análisis de texto.

Finalmente, la hipótesis H_0 formulada dentro del capítulo 1 puede ser aceptada, pues la evidencia mostrada en la tabulación de datos dentro del proceso de análisis

de texto sustenta en valores numéricos que el algoritmo Simple K-meas indica más del 50% de efectividad dentro de los porcentajes de error aceptados.

5.2. Recomendaciones

Planificar a tiempo un producto software en función de los recursos disponibles, las actividades a realizar, entre otros, tenderán a garantizar un producto de calidad.

Mejorar la calidad de la información recolectada dentro del proceso de extracción de texto, realizando una limpieza previa antes de continuar con el análisis de los datos.

Diseñar las interfaces de usuario permite la distribución correcta de la información dentro de la página web, logrando la aceptación del prototipo. En este contexto se recomienda utilizar la herramienta “axure rp” en su periodo de pruebas, pues ésta posee componentes que facilitan la navegación de la página web.

Investigar nuevas técnicas y algoritmos de aprendizaje no supervisado para disminuir el porcentaje de error al momento de realizar en análisis de texto.

Se recomienda una investigación previa de las utilidades incluidas dentro de la librería lucene, ya que sus componentes pueden llegar a solventar ciertas dificultades durante el proceso de desarrollo.

Seleccionar la técnica de aprendizaje no supervisado, se facilitó por la disponibilidad de la herramienta Weka, misma que incorpora varias técnicas y posibilita escoger aquella que da el menor margen de error.

5.3. Líneas de trabajos futuros

El trabajo realizado en esta tesis abre paso a nuevas líneas de investigación relacionadas con mejoras en el prototipo y la consideración de nuevos métodos para el proceso del desarrollo.

Dentro de la investigación se presenta algunas técnicas de análisis de datos, que involucran las características y beneficios de la clusterización. Puede ser

interesante predecir el éxito o fracaso de composiciones musicales utilizando nuevas técnicas de aprendizaje.

Respecto a la selección de atributos en el proceso del análisis de contenido, se consideró la relación entre la frecuencia y el texto de cada género musical. Puede ser interesante aumentar la cantidad de atributos para observar si el porcentaje de error arrojado por la herramienta Weka disminuye o incrementa.

Independientemente de la herramienta tecnológica que se utilice, la propuesta podría ampliarse al realizar una depuración sintáctico-semántica más rigurosa de los textos procesados.

BIBLIOGRAFIA

- Abdeen, M., & Tolba, M. (2010). Challenges and design issues of an Arabic web crawler. *Computer Engineering and Systems (ICCES)* (págs. 203-206). Cairo: IEEE.
- Abello, J., Pardalos, P., & Resend, M. (2002). Manual de conjuntos de datos masivos. En M. Najork, & A. Heydon, *High-performance web crawling* (págs. 22-45). Springer, Boston, MA.
- Álvarez Díaz, M. (diciembre de 2007). *Arquitectura para Crawling dirigido de información Contenida en la Web Oculta*. Coruña, España.
- Álvarez García, A., de las Heras del Dedo, R., & Lasa Gómez, C. (2012). *Métodos ágiles y scrum. Manual imprescindible*. Malaga-España: Anaya Multimedia.
- Baccigalupo, C., & Fields, B. (2009). *ismir*. Obtenido de <http://ismir2009.ismir.net/>
- Benítez Andrades, J. A. (diciembre de 2010). Resumen Tema 2: Crawling.
- Berenzweig, A., Logan, B., P.W. Ellis, D., & Whitman, B. (2003). *Proceedings International Conference on Music Information Retrieval (ISMIR)*. Obtenido de A large-scale evaluation of acoustic and subjective music similarity measures.
- Blacking, J. (1992). "The Biology of Music-Making," en *Ethnomusicology: An Introduction*. Norton.
- Boldi, P., Codenotti, B., Santini, M., & Vigna, S. (s.f.). UbiCrawler: A Scalable Fully Distributed Web Crawler.
- Bragado, D. M. (14 de marzo de 2016). Aplicación de técnicas de aprendizaje no supervisado para la clusterización temática de la red TOR. Alcalá.
- Bueno Crespo, A. (2013). Aprendizaje máquina multitarea mediante edición de datos y algoritmos de aprendizaje extremo. *dialnet.unirioja.es*.
- Cáceres Tello, J. (2006). Reconocimiento de patrones y el aprendizaje no supervisado. Alcalá.

- Cadavid Rengifo, H., & Gómez Perdomo, J. (diciembre de 2009). Sistema de extracción de cuerpos de texto de la web para tareas lingüísticas. *Revista ingeniería e investigación*, 54-60.
- Camargo Sarmiento, F. I., & Ordóñez Salinas, S. (2013). Evolución y tendencias actuales de los Web crawlers. *redalyc*, 19-35.
- Casado Valverde, Á. (2013). Sistema de extracción de entidades y análisis de opiniones en contenidos Web generados por usuarios. 58.
- Castillo, C. (2004). Web Crawling Eficaz. Chile.
- Chakrabarti, S., Van den Berg, M., & Dom, B. (1999). Focused crawling: a new approach to topic-specific Web resource discovery. En *Computer Networks* (págs. 1623-1640).
- Demicheri, S., & López, M. (2009). Aprendiendo por interacción en entornos estocásticos: análisis de performance para algoritmos de aprendizaje on y off policy. Villa María, Cordoba.
- echonest. (s.f.). Obtenido de <http://the.echonest.com/>
- Edwards, J., McCurley, K., & Tomlin, J. (s.f.). An Adaptive Model for Optimizing Performance of an Incremental Web Crawler.
- Eloffson, J. (13 de noviembre de 2013). Por qué los suecos componen canciones "perfectas". (B. Mundo, Entrevistador)
- Gallardo Campos, M. (2009). Aplicación de técnicas de clustering para la mejora del aprendizaje. Laganés.
- García Cambroner, C., & Gómez Moreno, I. (2009). Algoritmos de aprendizaje. *Universidad Carlos III de Madrid*.
- Garner, S. R. (s.f.). WEKA: The Waikato Environment for Knowledge Analysis. Hamilton, Nueva Zelanda.
- Gómez Flores, W. (s.f.). Análisis de Datos-Introducción al Aprendizaje Supervisado.

- Grande Benito, P. (24 de abril de 2008). Extracción de información con clasificación supervisada.
- Gupta, P., & Johari, K. (2009). Implementation of Web Crawler. *Emerging Trends in Engineering* (págs. 838-843). India: 2nd International Conference.
- Hornik, K., Buchta, C., & Zeileis, A. (14 de mayo de 2008). Open-source machine learning: R meets Weka.
- IBM. (s.f.). *IBM®*. Obtenido de https://www.ibm.com/support/knowledgecenter/es/SSPT3X_4.1.0/com.ibm.swg.im.infosphere.biginsights.text.doc/iewt_getstarted.html
- Ibrahim, S. N., Selamat, A., & Selamat, M. H. (2008). Scalable e-business social network using MultiCrawler agent. *Computer and Communication Engineering*, (págs. 702-706).
- Inacap. (s.f.). Obtenido de www.inacap.com
- Itzcoalt Alvarez, M. (s.f.). Desarrollo Ágil con Scrum. 35. Colombia.
- Jakob , L., & Nikola. (2015). *marsyas*. Obtenido de <http://marsyas.info/>
- Knees, P., Schedl, M., & Widmer, G. (2005). Multiple lyrics alignment: automatic retrieval of song lyrics. Austria.
- Lynn Wiener, J., Heydon, C., & Najork, M. A. (2001). *Estados Unidos Patente n° US6263364 B1*.
- Marx, K. (1959). *The German Ideology: Part 1*. New York: Norton.
- Matich, D. J. (marzo de 2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*.
- Mayer, R., Neumayer , R., & Rauber, A. (2008). Rhyme and style features for musical genre classification by song lyrics. Viena, Australia.

- Moreno, P. P., Puente, J., Pino Díez, R., & de la Fuente, D. (Noviembre-Diciembre de 2002). Utilización de las Redes Neuronales en la toma de decisiones: aplicación en un problema de secuenciación. Oviedo.
- Multi. (27 de marzo de 2013). *facilware*. Obtenido de <http://www.facilware.com/la-evolucion-de-la-web-1-0-2-0-y-3-0.html>
- Orjuela Duarte, A., & Rojas, M. (2008). Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. *Red de Revistas Científicas de América Latina y el Caribe*, 160-172.
- Ormeño Silva, E. V. (abril de 2014). Impletación y evaluación de algoritmos de minería de datos sobre hadoop mapreduce. Santiago, Chile.
- Prieto Álvarez, V. (2013). Arquitectura optimizada para un motor de búsqueda web eficiente. Coruña, España.
- Procesos de Software*. (s.f.).
- Qiang, Z. (2007). An Algorithm OFC for the Focused Web Crawler. *Machine Learning and Cybernetics*, (págs. 4059-4063).
- Qureshi, M. A., Younus, A., & Rojas, F. (2010). Analyzing the Web Crawler as a Feed Forward Engine for an Efficient Solution to the Search Problem in the Minimum Amount of Time through a Distributed Framework. *Information Science and Applications (ICISA)*, (págs. 1-8).
- Radhakishan, V., Farook, Y., & Selvakumar, S. (20 de julio de 2013). *ACM SIGSOFT*. Obtenido de Notas de Ingeniería de Software: <https://dl.acm.org/citation.cfm?id=1811236>
- Ringe, S., Francis, N., & Altaf, P. (septiembre de 2012). *International Journal of Computer Applications in Engineering Sciences*. Obtenido de <http://www.caesjournals.org/uploads/IJCAES-CSE-2012-073.pdf>

- Rui, H., Fen, L., & Zhongzhi, S. (2008). Focused Crawling with Heterogeneous Semantic Information. *Web Intelligence and Intelligent Agent Technology*, (págs. 525-531).
- Rungsawang, A., & Angkawattanawit, N. (7 de enero de 2004). Learnable topic-specific web crawler. Thailand.
- Sánchez-Montañés, M., Luis Lago, L., & González, A. (s.f.). *Escuela Politécnica Superior Universidad Autónoma de Madrid*. Obtenido de Métodos avanzados de aprendizaje artificial:
http://arantxa.ii.uam.es/~msanchez/docencia/maaa/transparencias/Intro_1112.pdf
- Schwaber, K., & Sutherland, J. (julio de 2013). La Guía Definitiva de Scrum: Las Reglas del Juego.
- Scrum Alliance. (s.f.). Obtenido de www.scrumalliance.org/why-scrum
- Shaojie, Q., Tianrui, L., Hong, L., Yan, Z., & Jing, P. (2010). SimRank: A Page Rank approach based on similarity measure. *Intelligent Systems and Knowledge Engineering (ISKE)*, (págs. 390-395).
- Sharma, N., Bajpai, A., & Litoriya, R. (2012). Comparison the various clustering algorithms of weka tools. *International Journal of Emerging Technology and Advanced Engineering*, 73-80.
- Soto, C., & Jiménez, C. (29 de agosto de 2011). Aprendizaje supervisado para la discriminación y clasificación difusa. Colombia.
- Tadapak, P., Suebchua, T., & Rungsawang, A. (2010). A Machine Learning Based Language Specific Web Site Crawler. *Network-Based Information Systems (NBIS)*, (págs. 155-161).
- The University of Waikato. (s.f.). Obtenido de <https://www.cs.waikato.ac.nz/ml/weka/>
- Ting, I. H., Hui-Ju, W., & Pei-Shan, C. (2009). Analyzing Multi-source Social Data for Extracting and Mining Social Networks. *Computational Science and Engineering*, (págs. 815-820).

- Tom, M. (julio de 2006). *The Discipline of Machine Learning*. Pittsburgh.
- Trigas Gallego, M., & Domingo Troncho, A. C. (2012). *google academico*. Obtenido de academia.edu:
https://s3.amazonaws.com/academia.edu.documents/39164786/mtrigasTFC0612memoria_1.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1515644578&Signature=aI5rmqcMIUFbaV8Cm7zCx8d7KyI%3D&response-content-disposition=inline%3B%20filename%3DMtrigas_TFC0612memoria
- Valdiviezo, P. M., Santos, O., & Boticario, J. (2010). Aplicación de métodos de diseño centrado en el uso y minería de datos para definir recomendaciones que promuevan el uso del foro en una experiencia virtual de aprendizaje. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal-Sistema de Información Científica*, 237-264.
- Vásquez Padilla, A. (enero de 2010). Sistema experto para la interpretación mamográfica (tesis). Mexico.
- Vela, P. (16 de junio de 2012). *sites.google*. Obtenido de <https://sites.google.com/site/crawlerwbri/arquitectura>
- Vidal Ruiz, E., & Casacuberta Nolla, F. (Septiembre de 2017). *Introducción al Aprendizaje Automático*. Valencia, España.
- Weinberger, B. (s.f.).
- Witten, I., & Eibe, F. (2005). *Data Mining*. Waikato: Morgan Kaufmann publications.
- Witten, I., Eibe, F., Trigg, L., Hall, M., Holmes, G., & Cunningham, S. J. (1999). *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*. Hamilton, Nueva Zelanda.
- Yuvarani, M., Iyengar, N., & Kannan, A. (s.f.). *LSCrawler: A Framework for an Enhanced Focused Web Crawler based on Link Semantics*. India.