



**Ministerio de Educación Superior
Universidad Oscar Lucero Moya, de Holguín
Facultad de Informática y Matemática**

Trabajo de Diploma en opción al Título de Ingeniero Informático

**SISTEMA DE CLASIFICACIÓN AUTOMÁTICA
DE NOTICIAS
A PUBLICAR EN EL PERIÓDICO ¡AHORA! DIGITAL**

Autor: Yisel Clavel Quintero
Tutores: M.Sc. Sergio Cleger Tamayo
Ing. María del Carmen Rodríguez Hernández

**Holguín, Cuba
Julio del 2010**

A mis padres

Agradecimientos

A mi familia, por ayudarme y estar ahí cada vez que los necesité...

A Robert, por escucharme, apoyarme y acompañarme en las innumerables noches de trabajo...

A mis amigas (o mejor, hermanitas) y amigos, por escucharme y aceptarme como soy y por los necesarios y frecuentes momentos de terapia y descanso...

A mis tutores. A mi tutora María del Carmen en especial, por guiarme, preocuparse y ocuparse de que la tesis se desarrollara satisfactoriamente y con la calidad requerida...

A mis profesores, por enseñarme casi todo lo que sé y prepararme para ser un profesional de éxito...

A la Revolución, por permitirme estar aquí y facilitarme los recursos para realizar esta tesis...

Y sobre todo, por todo el esfuerzo y dedicación, a mí.

*“El futuro de nuestra patria tiene que ser necesariamente un futuro
de hombres de ciencia.”*

Fidel Castro Ruz

Resumen

El periódico *jahora!* digital de la Casa Editora del mismo nombre, de la provincia de Holguín, se actualiza cuatro veces al día. La información que contiene el Sitio Web se divide en las noticias internas, confeccionadas por los periodistas de la entidad, y las externas, extraídas de fuentes periodísticas cubanas disponibles en Internet, que, antes de su publicación, son clasificadas manualmente por la Editora Web en las categorías definidas por el periódico. Su clasificación puede resultar ambigua, agobiante y en la mayoría de los casos retrasar el proceso editorial.

La presente investigación propone una solución a esta dificultad, a partir del diseño e implementación de un sistema informático capaz de clasificar automáticamente las noticias a publicar en el periódico *jahora!* digital con un alto grado de exactitud, confiabilidad y eficacia.

El documento hace un recorrido por todo el proceso ingenieril en torno al proyecto y toca puntos cardinales, como el estudio teórico, las fases detalladas de la metodología de desarrollo ICONIX, expresada a través del Lenguaje Unificado de Modelado y el estudio de sostenibilidad realizado.

Abstract

The digital journal *jahora!* of the Publishing House with the same name, located in the municipality of Holguín, is actualized four times every day. The information contained in this Web site is divided in internal news, which are developed by the entity journalists and external news, extracted from Cuban journalistic sources available in Internet. The news are classified manually by the Web Editor in the categories defined by the digital journal. This task may be ambiguous, exhausting and in most of the cases retard the editorial process.

This research proposes a solution to this difficulty by designing a computer system capable of automatically classify the news to publish in the digital journal, with a high degree of accuracy, reliability and efficacy.

The document makes a assessment of the whole engineering process regarding the project, thus bringing up cardinal subjects such as the theoretical study, detailed stages of the ICONIX methodology, expressed through the Unified Modeling Language, as well as the sustainability study carried out.

Índice de Contenido

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	8
1.1 INFORMATIZACIÓN DE LA PRENSA CUBANA	8
1.2 DESCRIPCIÓN DEL PERIÓDICO ¡AHORA! DIGITAL	9
1.2.1 Proceso Editorial	10
1.2.2 Estructura del Sitio Web ¡ahora!	12
1.3 REPRESENTACIÓN TEXTUAL	13
1.3.1 Transformación del corpus textual	15
1.3.2 Extracción de términos	15
1.3.2.1 Método Boolean	16
1.3.2.2 Método de Frecuencia	16
1.3.3 Reducción de la dimensionalidad.....	17
1.3.3.1 Técnicas basadas en selección de rasgos	17
1.3.3.2 Técnicas basadas en reparametrización	19
1.4 CLASIFICACIÓN AUTOMÁTICA DE TEXTOS	22
1.4.1 Tipos de clasificación	24
1.4.2 Algoritmos de clasificación	25
1.4.2.1 Modelo de Lenguaje Dinámico.....	25
1.4.2.2 Regresión Logística	26
1.4.2.3 Naive Bayes	28
1.4.2.4 TF/IDF	31
1.4.2.5 K-NN	33
1.5 EVALUACIÓN DE LA CLASIFICACIÓN	34
1.5.1 Corpus textual	34
1.5.2 Medidas a utilizar.....	35
1.5.3 Resultados de la evaluación	37
1.5.4 Estudio experimental	39
1.6 DESCRIPCIÓN DE LAS TECNOLOGÍAS PARA LA CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA ..	41
1.6.1 Herramientas para preprocesar y clasificar documentos	43
1.6.1.1 Lemur	44
1.6.1.2 Indri	45
1.6.1.3 Weka.....	46
1.6.1.4 Lucene.....	46
1.6.1.5 LingPipe.....	47
1.6.2 Herramientas de desarrollo	48
1.6.2.1 Eclipse.....	48
1.6.2.2 NetBeans	49
1.7 FUNDAMENTACIÓN DE LA METODOLOGÍA DE DESARROLLO UTILIZADA	50
1.7.1 ICONIX	52
CONCLUSIONES DEL CAPÍTULO	53
CAPÍTULO 2: DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	54
2.1 PLANIFICACIÓN	54
2.2 DEFINICIÓN DE REQUERIMIENTOS	55
2.2.1 Requerimientos funcionales	55
2.2.2 Modelo del Dominio	55
2.2.3 Modelo de Casos de Uso	57
2.2.3.1 Descripción de Casos de Uso.....	59
2.2.4 Requerimientos No Funcionales	59
2.3 ANÁLISIS, DISEÑO CONCEPTUAL Y ARQUITECTURA TÉCNICA	62
2.3.1 Análisis de Robustez	62
2.3.2 Arquitectura Técnica	63

2.3.2.1	<i>Arquitectura de capas</i>	64
2.3.2.2	<i>Modelo de Despliegue</i>	65
2.4	DISEÑO E IMPLEMENTACIÓN	65
2.4.1	Diagramas de Secuencia	65
2.4.2	Modelo de Clases	66
2.4.3	Estándar de Código	67
2.5	PRUEBA	67
2.6	VALORACIÓN DE SOSTENIBILIDAD	67
2.6.1	Dimensión Administrativa	68
2.6.2	Dimensión Socio-Humanista.....	72
2.6.3	Dimensión Ambiental	74
2.6.4	Dimensión Tecnológica	75
	CONCLUSIONES DEL CAPÍTULO	75
	CONCLUSIONES.....	77
	RECOMENDACIONES	78
	GLOSARIO DE TÉRMINOS.....	79
	BIBLIOGRAFÍA	81
	ANEXOS	I
ANEXO 1.	LISTA DE PALABRAS VACÍAS O <i>STOP WORDS</i> UTILIZADA	I
ANEXO 2.	MEDIDAS DE COMPARACIÓN ENTRE LOS ALGORITMOS DE CLASIFICACIÓN.	III
ANEXO 3.	COMPARACIÓN ENTRE HERRAMIENTAS DE RI.	IV
ANEXO 4.	DIAGRAMA DE GANTT.	V
ANEXO 5.	DESCRIPCIÓN DE CASOS DE USO.	VII
ANEXO 6.	DIAGRAMA DE ROBUSTEZ DEL PAQUETE SEGURIDAD.	XI
ANEXO 7.	DIAGRAMAS DE ROBUSTEZ DEL PAQUETE CLASIFICACIÓN.	XII
ANEXO 8.	DIAGRAMAS DE ROBUSTEZ DEL PAQUETE SERVICIO.	XVII
ANEXO 9.	DIAGRAMAS DE SECUENCIA DEL PAQUETE SEGURIDAD.	XIX
ANEXO 10.	DIAGRAMAS DE SECUENCIA DEL PAQUETE CLASIFICACIÓN.	XXI
ANEXO 11.	DIAGRAMAS DE SECUENCIA DEL PAQUETE SERVICIO.	XXVII
ANEXO 12.	DIAGRAMA DE CLASES.....	XXX
ANEXO 13.	ESTÁNDAR DE CÓDIGO UTILIZADO.	XXXIV
ANEXO 14.	CARACTERÍSTICAS DEL SISTEMA. COCOMO II.	XL

Índice de Tablas

Tabla 1.1:	Categorías y documentos de entrenamiento utilizados.	34
Tabla 1.2:	Promedio de Precisión y Cobertura.	35
Tabla 1.3:	Matriz de Confusión binaria.	36
Tabla 1.4:	Ranking según el <i>accuracy</i>	40
Tabla 1.5:	Test de Friedman.	40
Tabla 1.6:	Test de Wilcoxon.	41
Tabla 2.1:	Puntos de función desajustados (UFP).	69
Tabla 2.2:	Cantidad de Líneas de Código Fuente.	69
Tabla 2.3:	Factores de Escala.	70
Tabla 2.4:	Multiplicadores de esfuerzo.	70
Tabla 2.5:	Constantes.	71
Tabla 2.6:	Esfuerzo, Tiempo de desarrollo y Costo.	71

Índice de Figuras

Figura 1.1:	Proceso general de la clasificación automática de textos.	24
Figura 1.2:	Proceso de Entrenamiento.	29
Figura 1.3:	Vectores de Categorías de Entrenamiento.	32
Figura 1.4:	Comparación entre los clasificadores teniendo en cuenta el <i>accuracy</i>	37
Figura 1.5:	Comparación entre clasificadores según las medidas Micro y Macro Promedio de <i>Recall</i> , <i>Precision</i> y <i>F</i>	38
Figura 1.6:	Comparación de clasificadores en cuanto al tiempo de ejecución.	39
Figura 1.7:	Fases de ICONIX.	52
Figura 2.1:	Diagrama del Modelo del Dominio.	56
Figura 2.2:	Diagrama de Paquetes de Casos de Uso.	57
Figura 2.3:	Diagrama de Casos de Uso Paquete Seguridad.	58
Figura 2.4:	Diagrama de Casos de Uso Paquete Clasificación.	58
Figura 2.5:	Diagrama de Casos de Uso Paquete Automatización.	59
Figura 2.6:	Diagrama de Robustez del Caso de Uso Clasificar Noticias Internas del paquete Clasificación.	63
Figura 2.7:	Arquitectura del sistema.	64
Figura 2.8:	Diagrama de despliegue.	65
Figura 2.9:	Diagrama de Secuencia del Caso de Uso Clasificar Noticias Internas del paquete Clasificación.	66

Introducción

La comunicación es una necesidad social que proviene desde el surgimiento del hombre en el mundo. Transmitir sus pensamientos, sentimientos y necesidades a los semejantes fue siempre parte de la supervivencia, expresados a través de gestos, sonidos y señales, y posteriormente representados en paredes o piedras a través de la pintura rupestre. Con el tiempo, las expresiones empleadas fueron ajustándose a necesidades específicas, hasta el nacimiento del lenguaje oral y más tarde el escrito.

En el transcurso de la historia han existido cambios en los medios utilizados para transmitir la información. A partir del siglo X, se comienza a utilizar el correo, principalmente por monarcas, iglesias y nobles. La invención de la imprenta a principios del siglo XV significó un gran salto en cuanto a la forma de presentar la información, lo que permitió que a principios del siglo XVII surgiera el primer periódico impreso con cierta periodicidad en Inglaterra. [2]

Con la Revolución Industrial del siglo XIX, surge el rodillo en 1819, las primeras láminas metálicas en 1829, la rotativa en 1845 y en 1911 se crea la primera impresora. Se pasa a una sociedad urbana y el trabajador se convierte en lector de diarios. Los periódicos asumen un rol social y la noticia comienza a responder a los intereses públicos más que a los individuales. [2]

Con los avances tecnológicos del siglo XX se modificó hasta la manera más primitiva de la comunicación que es la escritura. En la década de los 80 del mencionado siglo, surgieron las bases de la moderna Internet y empezaron a expandirse por todo el mundo. Una década después se introdujo la *World Wide Web*, que se hizo común fácilmente y permitió la expansión de Internet, convirtiéndose ésta en la herramienta fundamental para la búsqueda de información. [5, 81]

“El fenómeno Internet no es más que la punta del iceberg, es el paradigma de la sociedad digital. Internet es un nuevo medio de comunicación. Primero fue la prensa, luego la radio, después la televisión. Ahora asistimos al nacimiento de un nuevo soporte para la información, que será la materia prima más valiosa del siglo

XXI. Son las redes de telecomunicaciones, que forman un nuevo espacio cultural y social. También es un nuevo territorio para la comunicación y el periodismo.” [30]

La expansión de Internet permitió el desarrollo de los medios de comunicación convencionales, los cuales han adoptado a la también llamada red de redes como otra vía mucho más económica para mostrar y recuperar información en cualquier parte del mundo, gracias a las características de multidireccionalidad, descentralización e interactividad de este nuevo medio; surgiendo así el periodismo de Internet.[30]

El periodismo que se efectúa en Internet recibe las más diversas denominaciones: digital, electrónico, Web, en red, telemático, online, incluso en línea.

En el ámbito periodístico, los medios de comunicación actúan en concordancia para la difusión de un suceso: la radio anuncia, la TV muestra y el diario publica. En cada uno, el tratamiento de la información es distinto y no se puede hablar de competencia. Sólo existe cuando se trata de obtener la primicia en idénticos medios.

La transmisión telemática, en lugar de la impresión en papel, hace que la prensa digital adquiera características propias que logran que se convierta en un nuevo medio. El periódico electrónico es un producto interactivo y multimedia, que integra diferentes recursos como el texto, la imagen, el vídeo y el sonido. [73]

Con la utilización de las redes telemáticas se pone a disposición del usuario del periódico *online* la lectura no secuencial, la inmediatez, la interactividad, la profundidad, la personalización y la actualización, entre otras características propias de Internet. Además, incluye chat, foros de debate, etc. [73]

El primer periódico del mundo en soporte electrónico se publicó en marzo de 1979, en Birmingham. Su nombre era “Viewtel 202” y se consideraba como un servicio complementario del diario Birmingham Post and Mail. Varias semanas después, en la ciudad de Norwich (Reino Unido), el rotativo Eastern Evening News, puso en marcha el “Eastel”, otro periódico de similares características.

A partir del año 1995 los periódicos digitales se expandieron por los diferentes continentes con el desarrollo de la Tecnología de la Información y las Comunicaciones (TIC). Cuba se ha ido insertando poco a poco en los espacios de

Internet. Primero nació el portal “Cuba”, con fines promocionales relacionados con el país. Precisamente uno de los puntos tratados en el Congreso de la Unión de Periodistas de Cuba (UPEC), efectuado en marzo de 1999, fue la importancia de difundir la verdad cubana a nivel internacional a través de la Web. Más adelante, se incorpora a la Red el periódico *Granma*, con elementos noticiosos del acontecer nacional en diversas esferas. Con la misma idea surge *Sierra Maestra*, como el primer periódico provincial en la Web.

En el 2000 se crean los periódicos digitales provinciales en cada territorio del país, entre ellos *AhoraWeb* en Holguín, para apoyar la Batalla de Ideas por el regreso del niño Elián González, como una de las numerosas actividades desarrolladas en función de la política trazada por el país para lograr satisfactoriamente su retorno, ya que era de vital importancia la opinión pública internacional y desmentir las tergiversaciones.

AhoraWeb sufrió transformaciones, como parte de la estrategia de la Casa Editora *¡ahora!*, de Holguín. Se comenzó a llamar *¡ahora!* y se destinó a difundir diariamente el acontecer de la provincia y la información más importante a nivel nacional e internacional.

La Casa Editora *¡ahora!* incluye, además, la revista *Ámbito*, destinada al intercambio cultural; la revista *Serranía*, dirigida a las zonas montañosas del Plan Turquino y el periódico *¡ahora!* impreso, órgano oficial del Partido Comunista de Cuba (PCC) en la provincia, con el fin de satisfacer las necesidades informativas relacionadas con las actividades políticas, económicas, sociales, culturales, deportivas y científicas.

El periódico *¡ahora!* digital se actualiza principalmente 4 veces al día. Debido al breve tiempo existente entre los horarios de actualizaciones del Sitio, la mayoría de las veces es insuficiente la cantidad de periodistas para cubrir todas las noticias del territorio. Además, la información a publicar requiere mayor atención en la redacción que las que se publican en el periódico impreso debido a la diversidad de lectores de nivel internacional, por lo que los términos empleados deben ser más explícitos y sencillos para facilitar la traducción.

Para la selección de las noticias a publicar la Jefa de Grupo del Sitio Web¹ tiene en cuenta las orientaciones del PCC, los trabajos periodísticos concluidos para el periódico impreso de la propia entidad y, las noticias publicadas en fuentes externas como: las agencias Prensa Latina (PL) y Agencia de Información Nacional (AIN) y restantes órganos de prensa del país como: Radio Angulo, Granma y Juventud Rebelde. Estas fuentes externas son consultadas a través del motor de búsqueda Google para completar las noticias requeridas en cada publicación.

Las noticias son clasificadas manualmente por las editoras antes de su publicación, cada una en una categoría, según la apreciación personal del tema a que éstas hacen referencia.

A partir de lo planteado anteriormente se detectaron algunas dificultades en la clasificación de noticias en el proceso de edición del periódico *¡ahora!* digital.

Muestra de esto es que:

- ✓ La clasificación de noticias se hace de forma manual, debido a la falta de una herramienta que la genere automáticamente.
- ✓ Resulta impreciso para la Editora Web decidir a cuál categoría pertenece la noticia a publicar.
- ✓ Las categorías en las que son clasificados los trabajos periodísticos son ambiguas y no desglosan completamente la información, lo que trae como consecuencia una difícil búsqueda y navegabilidad de los usuarios en el Sitio.
- ✓ La clasificación de un gran número de noticias al día provoca agotamiento visual en la Editora Web, lo que impide que fluyan favorablemente sus labores productivas.

A partir de las cuestiones antes mencionadas y el estudio de la gestión de la información en el proceso editorial del periódico *¡ahora!* electrónico, surge el siguiente **problema científico**: ¿Cómo favorecer la clasificación de noticias en el proceso de edición del periódico *¡ahora!* digital en su Casa Editora?

¹ Puede ocupar además, el rol de la Editora Web.

A partir del problema se definió el **objeto de investigación**: La clasificación de noticias en el proceso de edición de periódicos digitales.

Para solucionar el problema se persigue el siguiente **objetivo**: Desarrollar un sistema informático que favorezca la clasificación de noticias en el proceso de edición del periódico *¡ahora!* digital en su Casa Editora.

El objetivo de la investigación delimita el siguiente **campo de acción**: Informatización de la clasificación de noticias en el proceso de edición del periódico *¡ahora!* digital en su Casa Editora.

Para guiar la investigación, se elaboraron las siguientes **preguntas científicas**:

1. ¿Cuáles son los fundamentos teóricos en cuanto a la clasificación automática de textos para favorecer la clasificación de noticias en el proceso editorial de periódicos digitales?
2. ¿Cuál es el estado real de desarrollo obtenido en la informatización de la clasificación de noticias en periódicos digitales cubanos?
3. ¿Cómo diseñar un sistema informático para favorecer la clasificación de noticias en el proceso de edición del periódico *¡ahora!* digital?
4. ¿Será factible y sostenible la solución que se propone?
5. ¿Cómo implementar el sistema informático para favorecer la clasificación de noticias en el proceso de edición del periódico *¡ahora!* digital?

Para dar respuesta a las preguntas científicas y cumplir el objetivo trazado, se realizaron las siguientes **tareas**:

1. Elaborar los fundamentos teóricos en cuanto a la clasificación automática de textos para favorecer la clasificación de noticias en el proceso editorial de periódicos digitales
2. Diagnosticar el estado real de desarrollo obtenido en la informatización de la clasificación de noticias en periódicos digitales cubanos
3. Realizar la valoración de sostenibilidad de la solución propuesta según las dimensiones administrativa, socio-humanista, ambiental y tecnológica
4. Analizar, diseñar e implementar un sistema informático para favorecer la clasificación de noticias en el proceso de edición del periódico *¡ahora!* digital

Para dar cumplimiento a las tareas planteadas se emplearon **métodos teóricos, empíricos y estadísticos** de investigación científica.

Métodos Teóricos

Análisis y síntesis: Se utilizó con el fin de analizar la información manejada durante el proceso de edición del periódico Web *jahora!*, elaborar los fundamentos teóricos, descomponer las necesidades en requerimientos del sistema y realizar la valoración de sostenibilidad de la solución propuesta.

Histórico y lógico: Permitió ordenar cronológicamente la historia del periódico y estudiar la gestión de la información en el proceso editorial del periódico *jahora!* digital. Posibilitó expresar, en forma teórica, la esencia del objeto y las necesidades existentes en la Casa Editora *jahora!*

Enfoque sistémico: Fue utilizado para identificar y descomponer el sistema en subsistemas, así como las relaciones entre ellos, lo que facilitó además, organizar el trabajo y la lógica del proceso editorial del periódico *jahora!* digital.

Modelación: Permitió representar de manera simplificada el flujo de trabajo del periódico *jahora!* digital y una mejor comprensión del proceso editorial, así como obtener un producto de mayor calidad.

Métodos Empíricos

Entrevista: Posibilitó obtener información sobre cómo era el flujo de la información entre los trabajadores a través de las diferentes fases del proceso de edición del periódico *jahora!* digital, cómo la procesaban y qué necesitaban. Facilitó la comunicación entre los usuarios y la desarrolladora. Se entrevistó al Administrador de la Web, a la Jefa de Grupo del Sitio, a la Editora Web y otros trabajadores de la entidad.

Revisión de documentos: Fue utilizado para la recopilación de la información acerca de la estructura y funcionamiento del Sitio Web del periódico *jahora!* digital, así como relacionados con el objeto de estudio. Se revisaron documentos como: Perfil Editorial de *jahora!*, noticias del periódico digital y otros relacionados con la clasificación de textos en general.

Observación: Se realizó en diferentes períodos y permitió obtener un conocimiento profundo sobre las labores rutinarias en las diferentes fases del

proceso de edición, así como el funcionamiento del Sitio Web del periódico *¡ahora!* digital. A partir de este conocimiento se estableció la problemática a investigar.

Métodos Estadísticos

Friedman: Fue empleado para probar si existen diferencias entre los algoritmos de clasificación estudiados en el capítulo 1.

Wilcoxon: Se utilizó para identificar las diferencias entre pares de algoritmos de clasificación estudiados en el capítulo 1.

La presente investigación consta de introducción, dos capítulos, conclusiones, recomendaciones, glosario de términos, bibliografía y anexos.

Capítulo 1: Fundamentación del tema. Se expresa un estudio bibliográfico actualizado con el propósito de dar cumplimiento al objetivo trazado, incluyendo una descripción del objeto de estudio de la investigación y del proceso editorial del periódico *¡ahora!* digital, las principales tendencias y tecnologías para la construcción de la solución propuesta y la fundamentación de la metodología de Ingeniería de Software empleada.

Capítulo 2: Descripción y construcción de la solución propuesta. Estudio de sostenibilidad. Se describe el dominio del problema, incluyendo los requerimientos funcionales y no funcionales que debe cumplir la aplicación propuesta como solución, así como el modelo de casos de uso, los diagramas de robustez y secuencia. Se realiza, además, un estudio de la sostenibilidad del sistema informático, según las dimensiones administrativa, socio-humanista, ambiental y tecnológica.

Capítulo 1: Fundamentación teórica

En este capítulo se expone un conjunto de información referente a los conceptos que rodean el objeto de estudio y la solución propuesta, como la gestión de noticias en el periódico *¡ahora!* digital, la representación de documentos en la Recuperación de Información (RI) y la clasificación automática de textos. Se describen, además, las tecnologías para la construcción de la solución propuesta, haciendo énfasis en sus características, ventajas y desventajas y, se identifican las que se proponen para el desarrollo de esta investigación. Finalmente se fundamenta la metodología de desarrollo utilizada.

1.1 Informatización de la prensa cubana

Los medios de prensa cubanos son muy diversos en su modo de actuar. Si bien sus producciones son similares, sus métodos han sido muy particulares. Éstos medios están inmersos en un proyecto de informatización a nivel nacional.

Cuando se les dio la orientación de publicar sus sitios en Internet, no se realizó un proceso de cambio en el cual participaran todas las disciplinas necesarias. Por lo general, cada medio le asignó la tarea a uno de sus periodistas y ellos la asumieron y lograron desarrollar los sitios.[107]

Más adelante algunos directivos de la prensa se dieron cuenta que necesitaban incluir informáticos en la creación y optimización de sus sitios. Poco a poco las ideas de sistemas de gestión de contenidos, flujos de trabajos informatizados fueron entrando en la prensa.[107]

Hay problemas que son comunes a los medios de prensa y que pueden resolverse mediante sistemas generales, siempre que se puedan introducir estándares en la prensa cubana, ya que, si bien es necesario un software bueno, hace falta un plan para garantizar su éxito en la prensa.[107]

El Programa de Informatización de la Prensa define como su alcance, en cuanto al desarrollo de software[107]:

- Archivo y recuperación de imágenes
- Archivo y recuperación de audiovisuales
- Archivo y recuperación de textos (noticias, guiones, etc.)

- Archivo y recuperación de colecciones (PDFs de los periódicos, etc.)
- Sistemas de flujos de trabajo
- Plataforma Web que permita la creación de sitios Web integrados con el resto del sistema

En el caso del archivo de textos (e hipertexto), el formato para representar estos documentos debe ser muy versátil. Además, se puede evaluar la posibilidad de incluir[107]:

- Clasificación automática de textos
- Recomendación de enlaces a otros textos
- Corrector ortográfico basado en los mismos textos del archivo

Los periódicos digitales cubanos utilizan para la presentación de su sitio Web Sistemas de Gestión de Contenido (CMS, por sus siglas en inglés, *Content Management System*), lo que les permite hacer uso de las facilidades de gestión de la información que éstos brindan. Algunos cuentan con Sistemas de Gestión de Información para los periódicos impresos y otras herramientas aisladas para realizar algunas de las tareas del proceso editorial. La clasificación automática de textos, actualmente se realiza de forma manual, siendo una de las problemáticas que tienen en común.

1.2 Descripción del periódico *jahora!* digital

El periódico *jahora!* digital (www.ahora.cu) es una Web periodística considerada órgano provincial, a cargo del periódico *jahora!*, de Holguín. Está acogida al Registro de Publicaciones Seriadas de la República de Cuba bajo el código ISSN 1607-6389 y soportada en el CMS Joomla, que brinda facilidades para la actualización de contenidos, la interactividad y mayores garantías de seguridad. [19]

Se inserta en los medios digitales holguineros asumiendo el reto de la multimedialidad, hipertextualidad e interactividad, propias de la comunicación en Internet. [19]

El periódico *¡ahora!* digital tiene como objetivo [19]:

- Ampliar la presencia de Cuba en espacios físicos y virtuales, a partir del tratamiento de temas de la realidad de Holguín en primer lugar, sin descuidar lo más relevante que ocurra en el país y el mundo. Toda la información que en él se publique estará marcada por los referentes socio-culturales y políticos que han caracterizado a la prensa revolucionaria.
- Priorizar los contenidos que reflejen el desarrollo social, económico, cultural, científico y político de la provincia desde una óptica que resalte los logros y la justicia del Sistema Socialista y la Revolución.
- Convertir la Web en la voz de Cuba y su pueblo para la participación en debates internacionales que así lo ameriten, a partir de la publicación de documentos y posiciones oficiales del país.

El periódico está dirigido a todos los usuarios que se interesen por lo que acontece en Cuba, la realidad del país y su visión de los acontecimientos internacionales desde la óptica de quienes viven en una provincia a más de 700 Km de la capital. Ante la constante tergiversación de lo que sucede en la Isla y las mentiras que difunden las transnacionales de la información, el público destino deberá encontrar una mirada objetiva, sustentada en mayores dosis de argumentación y contextualización rigurosa. [19]

1.2.1 Proceso Editorial

Para la edición del periódico se tiene en cuenta el plan editorial emitido anualmente por el Comité Central, las orientaciones mensuales suministradas por el PCC provincial, las prioridades fijadas por el Director semanalmente a partir de las líneas y políticas editoriales trazadas y las iniciativas de los periodistas. Se seleccionan algunos de los trabajos periodísticos concluidos para el periódico impreso que sale con frecuencia semanal, así como noticias extraídas de fuentes externas cubanas para satisfacer las necesidades del periódico digital.

El proceso de edición comienza cuando la Jefa de Grupo del Sitio Web asigna los trabajos a los periodistas para luego ser revisados y publicados por la misma o por

la Editora Web. Si la noticia necesita estar acompañada por imágenes se les asigna la tarea a los fotógrafos de la entidad.

Las noticias internas son elaboradas por los periodistas a través del editor de texto *Microsoft Office Word* y luego son copiadas a la interfaz del Sitio correspondiente a la sesión de cada uno de ellos. Posteriormente le llega una notificación a la Editora Web de la nueva noticia realizada por el periodista, la cual es revisada, clasificada manualmente en una de las categorías predeterminadas² y finalmente publicada en el periódico *jahora!* digital.

Las noticias a publicar en el periódico digital requieren mayor atención en la redacción que las que se publican en el periódico impreso debido a la diversidad de lectores de otros países, por lo que los términos empleados deben ser más explícitos y sencillos.

La Web *ahora.cu* se actualiza en diversos momentos del día: a las 8:00 am, entre las 10:30 am y 1:30 pm por ser el horario de más tráfico en la página, a las 4:00 pm y por último entre las 10:30 pm y 11:30 pm por la diferencia de horario.

Debido al breve tiempo entre los horarios de actualización los periodistas no pueden cubrir todas las noticias del territorio, por lo que, a pesar de querer publicar más información de Holguín que nacional e internacional por el carácter provincial del periódico, no siempre se da cumplimiento a este objetivo. Además, existen muchas noticias nacionales e internacionales que por su importancia política y siguiendo con las orientaciones del PCC y las líneas de la entidad, es necesario no dejar de publicarlas.

Por todo lo antes expuesto, la Editora Web tiene en cuenta las noticias de otras fuentes periodísticas cubanas para la elección de las noticias nacionales e internacionales. Este proceso se realiza manualmente, buscando las noticias de interés disponibles en Internet, a través del motor de búsqueda Google, y copiándolas a la interfaz de edición del periódico proporcionada por Joomla. Las

² Son consideradas categorías: Holguín, Nacional, Internacional, Cultura, Deporte, Salud, Especiales, Opinión.

consultas más frecuentes realizadas en este buscador giran alrededor de: Cuba+Holguín, Cuba y Holguín.

Entre las fuentes externas más utilizadas se encuentran las que se listan a continuación por orden de prioridad, no obstante cualquier publicación de Cuba es una fuente potencial para el Sitio, ya sea un periódico, televisora o emisora del país:

- Prensa Latina (www.prensa-latina.cu)
- Agencia de Información Nacional (www.ain.cu)
- Granma (www.granma.cubaWeb.cu)
- Cuba Debate (www.cubadebate.cu)
- Radio Angulo (www.radioangulo.cu)
- Al día (www.aldia.cu)
- Juventud Rebelde (www.juventudrebelde.cu)
- Radio Habana Cuba (www.radiohc.cu)
- Cubahora (www.cubahora.cu)

Las noticias seleccionadas de PL y AIN, a diferencia de las elaboradas previamente por los periodistas y las de otros medios de prensa, se publican fielmente como están en la Web, dando crédito a la fuente original, es decir, no requieren de revisión ni tratamiento por el grado de confiabilidad de estas fuentes. Aunque puede darse el caso que alguna noticia de otra fuente no requiera tratamiento alguno.

1.2.2 Estructura del Sitio Web ¡ahora!

La Web *ahora.cu* se divide en varias secciones, término que se refiere a los espacios dedicados tanto a informaciones como a servicios de interés para el público según criterios de “cobertura informativa” y espacios de opinión, que por sus características, trascienden el valor informativo. Éstas enfatizan el carácter dialógico e interactivo de la publicación. [19]

ahora.cu utiliza canales RSS (*Really Simple Syndication*, por sus términos en inglés)³ para la comunicación de las noticias, servicio extendido en Internet. Es un sitio interactivo y ofrece diversos medios multimedia; emplea hipertexto; posee la opción de descargar ediciones publicadas del semanario *¡ahora!* en formato PDF; permite el acceso a las páginas personales de los periodistas, así como a galerías de imágenes.

Los lectores pueden inscribirse a la página, lo que les permite elaborar comentarios, interactuar en foros y usar el correo electrónico. Los no suscritos, además de poder visualizar la información contemplada en el Sitio, solamente pueden visualizar el foro y los comentarios reflejados en la página. Los datos almacenados de los usuarios se utilizan con fines de seguridad, y se contiene además, información sobre las noticias accedidas en las visitas realizadas.

Al actualizar el periódico no todas las noticias son reemplazadas de la portada, las de mayor relevancia permanecen un intervalo mayor de tiempo aunque su prioridad jerárquica cambia. En el caso contrario, las sustituidas no dejan de existir sino que pasan a ser miembros de las secciones asignadas.

La Base de Datos (BD) del Sitio, desde su concepción hasta hoy, radica en el Comité Central por política de seguridad, al igual que las de los demás periódicos electrónicos del país.

1.3 Representación textual

La representación textual no es más que mostrar el contenido recuperado de la Web o especificado por el usuario. Para representar documentos de lenguaje natural son muy utilizados, en la comunidad de minería de datos, *n*-gramas (del inglés *n-grams*) y el Modelo Espacio-Vectorial (VSM, por sus siglas en inglés, *Vector Space Model*), conocido también como bolsa o lista de palabras (del inglés *bag of words*).

³ Canal de noticias en formato XML que permite publicar artículos simultáneamente en diferentes medios a través de una fuente a la que pertenece.

El uso de n-gramas permite representar el texto mediante secuencias de palabras de longitud máxima n , lo que posibilita utilizar la información sobre la posición de la palabra en el texto. Facilita un mejor tratamiento de las frases negativas como "...excepto..." o "...pero no..." que de otra forma tomarían como relevantes las palabras que les siguen. [31]

En el VSM cada documento se representa como un vector de dimensión j , siendo j el número de palabras y en donde cada palabra constituye una componente del vector y representa una característica, la cual puede ser *booleana* (aparece o no en el documento) o basada en frecuencias (el número de veces que ha aparecido en el documento). Esta representación ignora el orden de aparición de las palabras en el texto y las relaciones semánticas y gramaticales entre ellas.

Un vector documento dado, en cada componente tiene un valor numérico para indicar su importancia. [14] Un documento d_j es representado como un vector de n dimensiones como sigue:

$$d_j = \langle W_{1j}, W_{2j}, \dots, W_{nj} \rangle$$

donde W_{ij} caracteriza la importancia de un término t_i en el documento d_j , es decir, los elementos del vector son pesos asignados a los términos a partir del vocabulario. El W_{ij} puede ser determinado a partir de uno de los métodos de frecuencia: Frecuencia del Término (TF, por sus siglas en inglés, *Term Frequency*); Frecuencia Inversa del Documento (IDF, por sus siglas en inglés, *Inverse Document Frequency*) o Frecuencia del Término / Frecuencia Inversa de Documento (TF/IDF). El método TF/IDF es el más comúnmente usado en el Modelo Espacio-Vectorial. [94]

Algunas palabras contienen más significados que otras (por ejemplo: los sustantivos o los verbos), por lo que para representar los documentos obtenidos es necesario realizar algunas tareas previas de procesamiento documental para seleccionar aquellas palabras que realmente son útiles para la recuperación. La transformación del corpus, la extracción de términos y la reducción de la dimensionalidad son algunas de las tareas de Procesamiento de Lenguaje Natural (del inglés *Natural Language Processing*).

1.3.1 Transformación del corpus textual

El objetivo de la transformación del corpus es convertir los ficheros de entrada en una secuencia de ítems lingüísticos (*tokens*⁴ de palabras). [14]

El primer paso en la transformación del corpus es reconocer los componentes textuales desde los diferentes formatos. Sobre la base de los componentes extraídos del texto, éste debe ser dividido en una secuencia de *tokens*. Luego, la secuencia resultante de *tokens* se transforma. Se puede convertir todas las letras a mayúsculas o todas a minúsculas; eliminar los signos de puntuación; omitir los *tokens* que contienen caracteres alfanuméricos; convertir los dígitos a un dígito predeterminado, a las palabras que los describen o eliminarlos; identificar o marcar los nombres de personas, localidades, organizaciones y productos; y sustituir las contracciones y abreviaturas por la expresión completa que representan. [14]

1.3.2 Extracción de términos

La extracción de términos “parte de una secuencia de *tokens* obtenida a partir de la transformación del corpus y produce una secuencia de términos indexados basados en esos *tokens*”. [14]

Representar documentos de lenguaje natural por el significado de un conjunto de índices de términos es un reto, sobre todo porque la información siempre depende del contexto. La forma en que la secuencia de términos indexados es usada depende del procesamiento posterior que se le vaya a realizar al documento y si el vocabulario ha sido construido o no. [14]

El proceso de asignar peso de un término a un documento es llamado indexación de documentos o pesado de término. [29] Los métodos tradicionales para la indexación de documentos son el método *Boolean* [94] y el método de Frecuencia [94].

⁴ Cadenas de caracteres delimitadas por espacios en blanco.

1.3.2.1 Método Boolean

Este método es el más simple para la representación de documentos. Cada peso de término es 0 o 1; 0 significa que el término no está presente en el documento y 1 significa que el término está presente en el documento. Se calcula como sigue:

$$w_{ij} = \begin{cases} 0 & \text{si } t_i \notin d_j \\ 1 & \text{en caso contrario} \end{cases}$$

1.3.2.2 Método de Frecuencia

Este método asigna valores no negativos a la matriz de peso, en vez de valores binarios como en el método *Boolean*. Según la frecuencia del término y la frecuencia del documento, hay tres métodos de indexación: TF; IDF y TF/IDF, la cual es la combinación de los dos primeros métodos.

Método TF

En el método TF los pesos de término son calculados basados en la frecuencia de los términos en el documento. Dado f_{ij} , que es el número de ocurrencias del término t_i en el documento d_j , el w_{ij} puede ser una función de f_{ij} .

$$w_{ij} = TF(f_{ij})$$

TF debe ser una función monótona (que no cambia) de f_{ij} , indicando que la importancia de un término es creciente cuando su frecuencia es alta; como sigue:

$$TF(f_{ij}) = \sqrt{f_{ij}} \text{ or } w_{ij} = 1 + \log_2(f_{ij}) \text{ or } w_{ij} = f_{ij}$$

Método IDF

En el método IDF se calcula la matriz de peso de término como sigue:

$$w_{ij} = \log_2 \frac{m}{df_i}$$

donde m es el número de documentos en la colección de documentos, df_i es la frecuencia del documento del término t_i , el número de documentos en el cual el término t_i ocurre. La razón de usar los pesos IDF es que los términos que ocurren

en muy pocos documentos son mejores discriminadores que los que ocurren en muchos documentos.

Método TF/IDF

El tercer método combina la frecuencia del término f_{ij} y la frecuencia del documento df_i . Es muy común y w_{ij} se define a partir de la multiplicación de los métodos TF e IDF.

$$w_{ij} = \begin{cases} TF(f_{ij}) \log_2 \frac{m}{df_i} & \text{si } f_{ij} \geq 1 \\ 0 & \text{si } f_{ij} = 0 \end{cases} \quad (1)$$

En este método, un término en un documento es caracterizado por la TF y la IDF. Éste es el más usado para la representación de documentos.

1.3.3 Reducción de la dimensionalidad

Es esencial controlar la dimensionalidad del espacio del vector documento para reducir el número de rasgos (términos) que son finalmente usados para representar los documentos. Las razones principales son:

- La complejidad de muchos algoritmos de agrupamiento y clasificación depende crucialmente del número de rasgos y reducirlo es necesario para hacer estos algoritmos tratables.
- Existen palabras que son irrelevantes y provocan la obtención de peores resultados, por tanto, eliminarlas puede realmente aumentar la eficiencia del agrupamiento o la clasificación a realizar. [14]

La reducción de la dimensionalidad abarca técnicas de procesamiento de documentos que controlan la dimensionalidad del vector: selección de rasgos [94] y reparametrización, o combinación de ambas técnicas.

1.3.3.1 Técnicas basadas en selección de rasgos

La implementación de estas técnicas tiene como entrada un conjunto de rasgos y como salida un subconjunto de esos rasgos, los cuales son relevantes.

Eliminación de palabras vacías

La eliminación de palabras de parada (del inglés *stop words*), gramaticales o vacías [111], es una de las técnicas de selección más utilizadas. Métodos de filtrado también se utilizan para decidir cuándo incluir un término en el vocabulario o no, tratando cada uno independientemente y evaluándolo. El vocabulario final se establece seleccionando todos aquellos rasgos que su puntuación sea superior o inferior a un umbral predeterminado, o los m mejores rasgos, es decir, los m rasgos con mayor o menor puntuación acorde a la magnitud de la puntuación. [14] Las *stop-words* son palabras muy frecuentes en el documento y no ofrecen información alguna sobre su contenido, como artículos, preposiciones, conjunciones y pronombres, que no son buenos discriminadores del contenido del documento y dependen del idioma y del dominio. [45]

Existe también un grupo grande de palabras que sólo ocurren una vez, que tampoco son útiles para la recuperación de la información. Las palabras con frecuencia media son las más descriptivas. [45]

Existen dos formas de identificar estas palabras para su posterior eliminación:

- Consultando una lista de palabras vacías para el lenguaje a tratar.
La lista de *stop-words* puede establecerse manualmente, lo que posibilita la eliminación solo de palabras gramaticales, pero tiene como desventaja que es dependiente del lenguaje. También se puede construir automáticamente basada en una colección de documentos bajo consideración; de esta forma la lista no es dependiente del lenguaje pero se pueden eliminar sustantivos, verbos y adjetivos.
- Identificando las palabras más o menos frecuentes que superan un cierto umbral previamente calculado.

Se eliminarían todos los términos cuyas frecuencias son:

- Superiores a un umbral predefinido (términos con frecuencia de aparición alta se asumen que son comunes y que no tienen poder discriminante).

- Inferiores a un umbral predefinido (términos que raramente aparecen en una colección de documentos tendrán poco poder discriminante y pueden ser eliminados).

Umbral de frecuencia de documentos

Para definir un umbral, el número de documentos en los cuales el término t aparece al menos una vez, se pueden excluir todos los términos del vocabulario cuya frecuencia de documentos es menor que el umbral, ya que los términos que ocurren en solo muy pocos documentos improbablemente llevan información que posibilite la clasificación o el agrupamiento, más bien tienden a provocar ruido.[14]

Frecuencia inversa de documentos y TF/IDF

La importancia de los términos se asume inversamente proporcional al número de documentos en los cuales el término particular aparece. Habiendo eliminado las palabras gramaticales, se puede asumir que la importancia de un término se incrementa con su frecuencia. Combinando estas ideas se formuló la medida TD/IDF, que se calcula con la ecuación (1) expuesta en el [subepígrafe 1.3.2.2](#). [14] Una combinación similar de la frecuencia del término y la frecuencia del documento inverso es usualmente usada para asignar pesos a los términos en la selección de rasgos. [94]

1.3.3.2 Técnicas basadas en reparametrización

Algunas de las técnicas lingüísticas que reducen la dimensionalidad por reparametrización son: la homogeneidad ortográfica (del inglés *spelling*), la segmentación (del inglés *stemming*), la lematización,[111] el Análisis de Latencia Semántico (LSA, por sus siglas en inglés, *Latent Semantic Analysis*)[58], así como el uso de tesauros y ontologías.

Los algoritmos basados en estas técnicas tienen como entrada un conjunto de rasgos derivados de la colección de documentos y como salida un nuevo conjunto que contiene menos rasgos. Los rasgos se forman a partir de combinaciones y

transformaciones de los existentes, manteniendo y en algunos casos mejorando el poder discriminante.

Homogeneidad ortográfica

Esta técnica permite convertir todas las palabras del léxico en un idioma estándar. Una de las aplicaciones más conocidas es en el idioma Inglés, a la hora de convertir las palabras del léxico norteamericano al británico o viceversa. Ejemplo: sustituir “capitalize” por “capitalise” y “colour” por “color”.

Stemming

Esta técnica permite reducir la dimensionalidad del espacio de rasgos y convierte las palabras a su base morfológica, es decir, extrayendo su raíz léxica (lexema), con el objetivo de obtener un único término a partir de diferentes palabras que constituyen esencialmente variaciones morfológicas con un mismo significado. [13] La raíz es la parte de la palabra que queda al eliminar afijos (prefijos, infijos y sufijos). Un mismo lexema representará a una familia de palabras semántica y morfológicamente relacionadas. [45]

Lematización

Este proceso es más preciso que la segmentación. Comprende la eliminación de los plurales, de las conjugaciones verbales y su reducción al infinitivo, la conversión de términos femeninos a masculinos, etc. No es dependiente del dominio pero es dependiente del lenguaje. [45] Por ejemplo:

- El término *niño* se obtiene a partir de *niños* y *niñita*.
- En el caso de los verbos se obtiene el infinitivo *amar* a partir de *amo* y *amará*.

Análisis de Latencia Semántico

El LSA tiene una alta complejidad computacional. Es una técnica estadística para descubrir automáticamente una estructura semántica, es decir, encontrar

asociaciones semánticas entre los términos en una colección de documentos. Resuelve el problema de sinónimos.

Uso de Tesauros

La palabra tesoro⁵ se refiere a un vocabulario controlado y dinámico de términos incluyendo relaciones semánticas y genéricas entre ellos, que se aplica a un dominio particular del conocimiento y es utilizado tanto en la indexación como en la RI para trasponer a un lenguaje más estricto y controlado, el lenguaje natural empleado en los documentos. [45]

El uso de tesauros depende del dominio y del lenguaje. Orienta a los indizadores y a los usuarios sobre los términos que pueden utilizar y así ayuda a mejorar la calidad de la RI. Se considera como un único término aquellos que sean sinónimos o cuasi-sinónimos⁶ y hace corresponder todos los términos indexados a las clases que ellos pertenecen. [27]

De forma general, un tesoro comprende lo siguiente:

1. Un listado de términos preferidos, que se los ordena en forma alfabética, temática y jerárquicamente.
2. Un listado de sinónimos de esos términos preferidos, llamados descriptores.
3. Una jerarquía o relaciones entre los términos.
4. Las definiciones de los términos, para facilitar la selección de los mismos por parte del usuario.
5. Un conjunto de reglas para usar el tesoro.

Se eligieron las técnicas de eliminación de palabras vacías (ver [Anexo 1](#)) y *stemming* para el preprocesamiento de los documentos en el sistema informático propuesto como solución; debido a que las palabras vacías no ofrecen ninguna información útil para la clasificación; y el *stemming*, a pesar de provocar pérdida de información, ésta no es significativa en el dominio del problema. Además,

⁵ Palabra derivada del neo latín que significa *tesoro*. Proviene etimológicamente del latín *thesaurus*, el cual tiene su origen del griego clásico *thesauros* (θησαυρός), que significa *almacén, tesorería*.

⁶ Términos equivalentes utilizados para la indexación que tienen diferentes significados en el lenguaje ordinario.

ambas técnicas permiten la reducción de la dimensionalidad de los documentos, aspecto muy importante para maximizar el rendimiento (minimizando el costo en tiempo) y la efectividad del algoritmo utilizado.

1.4 Clasificación automática de textos

Debido a la expansión que Internet ha experimentado en todo el mundo, cada vez es mayor el número de fuentes de contenidos y el volumen de datos que se tiene al alcance. Este crecimiento explosivo de documentos disponibles complica su exploración y análisis. Por consiguiente, son necesarios nuevos métodos que ayuden a los usuarios a filtrar y estructurar la información relevante. Por ello, poder organizar la información de forma automática ha pasado a ser una tarea de vital importancia, y llevar a cabo una gestión eficiente de la información se ha convertido en algo imprescindible. [68]

En la minería de texto se hace uso del Procesamiento de Lenguaje Natural, tratado en el [epígrafe 1.3](#), y del Aprendizaje Automático (ML, por sus siglas en inglés, *Machine Learning*).

El **Aprendizaje Automático** se puede definir como el proceso de encontrar la hipótesis más probable dado el conjunto de ejemplos de entrenamiento y un conocimiento a priori sobre la probabilidad de cada hipótesis. [1]

Dentro de esta técnica se encuentra el **Aprendizaje Supervisado** (ejemplo: clasificación de documentos), que se define como descubrimiento de patrones en los datos que relacionan atributos con un atributo objetivo. [44] En la clasificación, el aprendizaje se “alimenta” de un conjunto de textos de entrenamiento que han sido clasificados manualmente de antemano, que sirven como ejemplo para construir un clasificador que después será utilizado para detectar a qué categoría pertenecen los nuevos documentos (ver Figura 1.1) [41]. En el **no Supervisado** (ejemplo: agrupamiento de documentos), no existen categorías o atributos objetivos previamente definidos y se realiza la exploración de los datos para encontrar la estructura intrínseca. [44]

La **clasificación o categorización automática de texto** consiste en un conjunto de algoritmos, técnicas y sistemas capaces de asignar un documento a una o

varias categorías o grupos de documentos, contruidos según su afinidad temática.

Para el diseño de un clasificador se pueden emplear diferentes técnicas de aprendizaje, debiendo disponer para ello de un conjunto de documentos (**conjunto de entrenamiento**), que previamente han sido clasificados dentro de una determinada categoría. Estos algoritmos de aprendizaje o entrenamiento requieren una representación estructurada de los documentos. La más empleada es la basada en el Modelo de Espacio-Vectorial, tratada en el [epígrafe 1.3](#).

Los textos no pueden ser directamente interpretados en primer lugar por el aprendiz, y posteriormente por el clasificador una vez que éste ha sido construido. Por lo tanto, es necesario realizar un procesamiento previo de los documentos (ver [epígrafe 1.3](#)), el cual necesita ser uniformemente aplicado al conjunto de entrenamiento, así como a los nuevos documentos a ser catalogados.

Una vez que el clasificador ha sido entrenado con el correspondiente grupo de textos, su efectividad se evalúa comparando las categorías que ha asignado a los documentos del conjunto de prueba con las que éstos ya tenían asignadas. Este algoritmo permite alcanzar una precisión comparable a la obtenida por expertos humanos.

En la siguiente figura se muestra una visión general de los elementos y procesos necesarios dentro de la clasificación automática de textos.

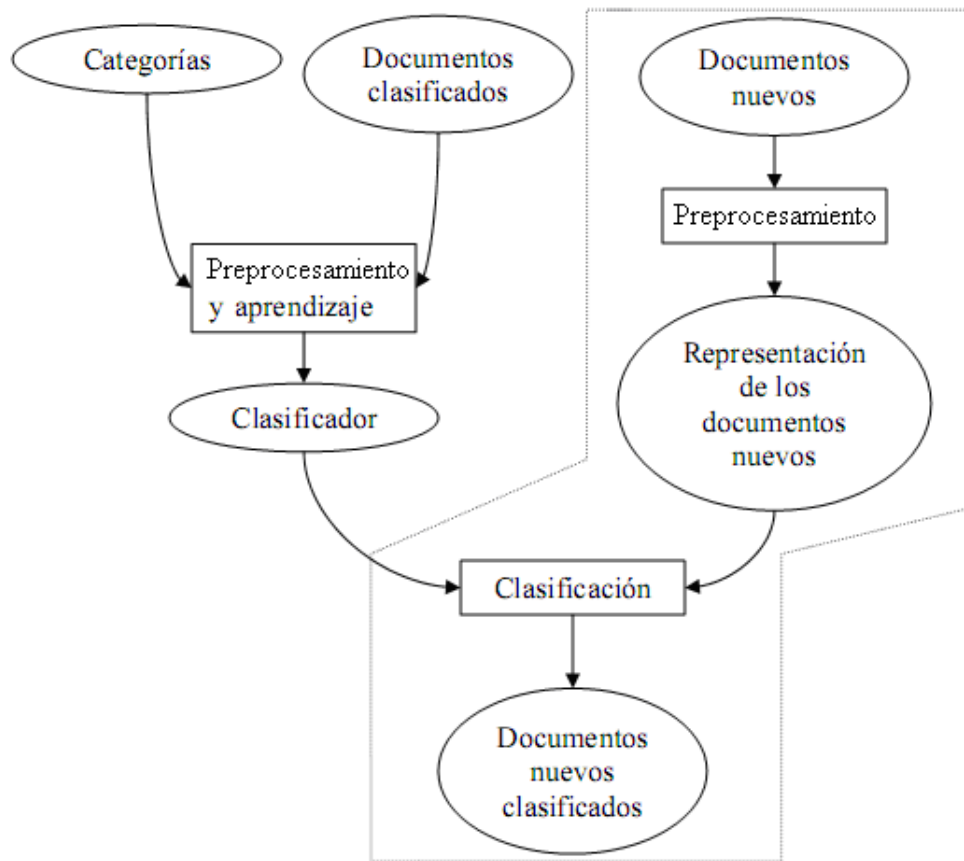


Figura 1.1: Proceso de clasificación automática de textos.

1.4.1 Tipos de clasificación

Existen varios tipos de clasificación como son:

- **Simple o Single-label:** Exactamente una categoría se asigna a cada documento. En ocasiones es llamada categorías no solapadas.
- **Múltiple o Multilabel:** Cualquier número de etiquetas o categorías se pueden asignar al mismo documento. En ocasiones es llamado categorías solapadas.
- **Binaria:** Tipo especial del primer tipo, donde un documento se asigna a una categoría o a su complementario (Por ejemplo: *spam*⁷ – *no spam*).

⁷ Correo basura o no solicitado, normalmente con contenido publicitario, que se envía de forma masiva.

El caso binario (por tanto el simple también) es más general que el múltiple, porque el problema de asignar varias etiquetas a un documento, se puede ver como varios problemas binarios, uno por categoría existente, evaluando si pertenece o no a dicha categoría. [96]

1.4.2 Algoritmos de clasificación

Hay tres componentes principales implicados en el proceso de clasificación de texto. Inicialmente está el conjunto de objetos para ser clasificados, que son los documentos textuales. El segundo componente son las clases finales predefinidas por los especialistas o expertos del dominio de la información a clasificar. El tercer componente es el algoritmo que actúa como clasificador. [31]

Existen gran cantidad de algoritmos propuestos para la clasificación de textos. La mayor parte de ellos no están preconcebidos con ese fin, sino que se han propuesto para tareas generales de clasificación y se han adaptado para su uso en la minería de texto. Entre ellos, se reconocen en la literatura como clásicos: *Naive Bayes* (Modelo multinomial y Modelo multivariado Bernoulli), *Rocchio*, Vecinos más cercanos (NN, por sus siglas en inglés, *Nearest Neighbour*), K-Vecinos más cercanos (K-NN, por sus siglas en inglés, *K-Nearest Neighbour*), Máquinas de Vectores de Soporte (SVM, por sus siglas en inglés, *Support Vector Machine*), ALgoritmos de VOTación (ALVOT), algoritmos basados en Árboles de Decisión o Redes Neuronales. Existen otros como: Modelo de Lenguaje Dinámico (del inglés *Dynamic Language Model*), Regresión Logística (del inglés *Logistic Regression*) y los basados en el TF/IDF. A continuación se analizarán algunos de estos.

1.4.2.1 Modelo de Lenguaje Dinámico

El Modelo de Lenguaje Dinámico, es un clasificador del modelo de lenguaje, que acepta eventos de entrenamiento de secuencias de caracteres categorizados. El entrenamiento se basa en un estimador multivariado para distribución de la categoría y los modelos de lenguaje dinámico para los estimadores de secuencia de caracteres por categoría.

Construye un clasificador dinámico de modelo de lenguaje sobre las categorías especificadas, usando modelos de proceso de carácter *n-gram* por categoría, creados a partir de la longitud máxima especificada del *n-gram* y un estimador global de la categoría.

La clasificación de un nuevo documento se basa en el logaritmo de las probabilidades *joint*. Es decir, se determina la probabilidad de que la secuencia de caracteres del nuevo documento pertenezca a la categoría especificada:

$$\log_2 P(s, cat) = \log_2 P(s|cat) + \log_2 P(cat)$$

donde $P(s|cat)$ es la probabilidad del carácter de secuencia cs en el modelo de lenguaje para la categoría cat y donde $P(cat)$ es la probabilidad asignada por la distribución multivariada sobre las categorías.

Los *scores* se definen, para ser muestras normalizadas de la siguiente manera:

$$score(s, cat) = \frac{\log_2 P(s|cat) + \log_2 P(cat)}{(sLongitud + 2)}$$

El valor de mayor probabilidad *joint* será determinado por:

$$ARGMAX_{cat} P(cat|cs) = \frac{P(s, cat)}{P(s)}$$

1.4.2.2 Regresión Logística

La Regresión Logística es un modelo probabilístico de clasificación, conocido además, como discriminación logística. Proporciona una clasificación multinomial, es decir, permite más de dos categorías posibles de la salida. La regresión logística multinomial es además conocida como *polytomous*, *polychotomous* o regresión logística multiclase.

Es uno de los modelos lineales generalizados más frecuentes, ya que modelizan una probabilidad. En este caso, la variable de respuesta tiene dos o más posibilidades, cada una con su respectiva probabilidad, siendo la suma de probabilidades igual a uno.

Los modelos de regresión logística se estiman de los datos del entrenamiento que consisten en una secuencia de vectores y la referencia a sus categorías. Los

vectores son arbitrarios, con sus dimensiones representando los rasgos de los objetos de entrada que son clasificados. Las categorías son discretas, y se deben numerar continuamente a partir de 0 hasta el número de categorías menos una.

Para un vector de entrada x , la probabilidad condicional de una categoría k se define como:

$$\begin{aligned} p(0|x) &= \frac{\exp(\beta_0^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \\ p(1|x) &= \frac{\exp(\beta_1^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \\ &\dots \\ p(k-2|x) &= \frac{\exp(\beta_{k-2}^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \\ p(k-1|x) &= \frac{\exp(\beta_{k-1}^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \end{aligned}$$

Normalizando a través de la suma se estima la probabilidad condicional como:

$$\begin{aligned} p(0|x) &= \frac{\exp(\beta_0^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \\ p(1|x) &= \frac{\exp(\beta_1^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \\ &\dots \\ p(k-2|x) &= \frac{\exp(\beta_{k-2}^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \\ p(k-1|x) &= \frac{\exp(\beta_{k-1}^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-2}^* x) + \exp(\beta_{k-1}^* x)} \end{aligned}$$

quedando de manera general, para $c < k-1$:

$$p(c|x) = \frac{\exp(\beta_c^* x)}{\exp(\beta_0^* x) + \dots + \exp(\beta_{k-1}^* x)} \quad (2)$$

y para $c = k-1$:

$$p(c|x) = \frac{1}{1 + \sum_{i \neq c} \exp(\beta_i x_i)}$$

El objetivo de generar un modelo de regresión logística con la ecuación (2), es estimar la probabilidad de que el documento x pertenezca a la categoría c . Para ello, se determina la probabilidad condicional del documento x para cada una de las categorías. De tal manera, de que por cada solución, se pueda conocer de manera ordenada (de mayor a menor) la probabilidad del documento a clasificar por cada categoría.

La decisión de asignar el documento a una de las categorías puede estar basado en la comparación de las probabilidades estimadas con un umbral, o más común, calcular qué decisión es la más óptima. [39]

La regresión logística es uno de los mejores clasificadores probabilísticos, a pesar de tener como principal inconveniente que es relativamente lento para entrenar, comparado con otros clasificadores, además de requerir extensivos ajustes en la manera como seleccionan los rasgos.

1.4.2.3 Naive Bayes

El clasificador Naive Bayes (NB) es un modelo probabilístico muy conocido y con una implementación muy simple. Existen dos modelos comunes para la clasificación de texto de NB, discutidos por McCallum y Nigam [69], Modelo Multinomial (del inglés *Multinomial Model*) y Modelo Multivariado Bernoulli (del inglés *Multivariate Bernoulli Model*). En ambos modelos la clasificación se basa en la teoría probabilística, en especial en el teorema de Bayes:

$$P(c_i | d_j) = \frac{P(c_i)P(d_j | c_i)}{P(d_j)} = P(c_i) \prod_j P(d_{j,i} | c_i)$$

donde:

$P(c_i | d_j)$: es la probabilidad de que d_j pertenezca a c_i

$P(d_j)$: es la probabilidad de que un documento seleccionado al azar tenga el vector d_j como su representación. Esta probabilidad es ignorada, ya que es constante.

$P(c_i)$: es la probabilidad de que un documento seleccionado aleatoriamente pertenezca a c_i .

$P(d_j|c_i)$: es la probabilidad de que el documento a clasificar aparezca en los documentos que pertenecen a la categoría en cuestión. Es el cálculo más costoso ya que hay muchos documentos.

El funcionamiento del algoritmo NB, básicamente, trata de estimar la probabilidad de que un documento pertenezca a una categoría.

Modelo Multinomial

En este modelo para calcular la probabilidad se utilizan *tokens* en lugar de *n-grams*. Es decir, Naive Bayes es aplicado al resultado de textos tokenizados en un modelo *bag of words* donde los *tokens* son asumidos para ser independientes uno de otros. Este modelo a menudo es llamado modelo de lenguaje unigrama (del inglés *unigram language model*).

El entrenamiento consiste en crear modelos de lenguaje tokenizado por cada categoría (ver Figura 1.2):

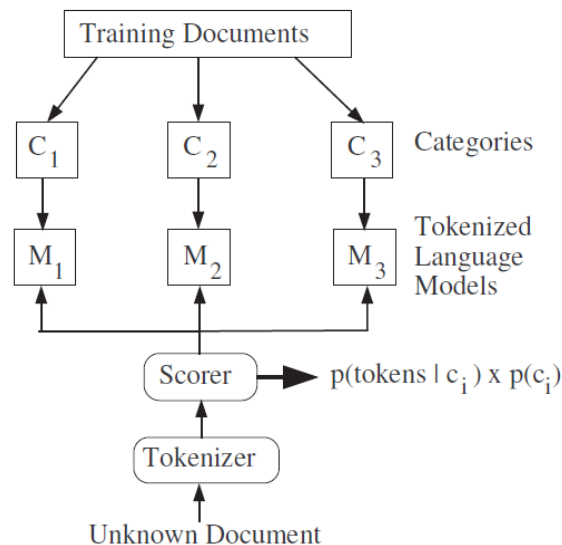


Figura 1.2: Proceso de Entrenamiento.

Un modelo de lenguaje tokenizado contiene las estimaciones de la probabilidad para los *tokens* w que han sido vistos en el texto de los documentos de entrenamiento para una categoría. La probabilidad de una categoría c se basa en la proporción del número de los documentos de entrenamiento asignados a c .

Estas dos probabilidades que se obtienen de los documentos de entrenamiento se utilizan para clasificar una nueva instancia de un documento d dentro de un conjunto finito c de clases predeterminadas. Esto significa que, dada una clase c y un conjunto de *tokens* w del nuevo documento a clasificar, se calcula la probabilidad de que dicho documento se clasifique dentro de la categoría c . En otras palabras, para una categoría c , es el producto de las probabilidades individuales de los *tokens* w dada la categoría y la probabilidad de la categoría por sí mismo:

$$P(c|w_j) = P(c) \prod_{i=1}^d P(w_i|c_j)$$

El de mayor probabilidad es identificado como la mejor categoría para el documento que está siendo clasificado:

$$MejorCategoría = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i=1}^d P(w_i|c_j)$$

Los *tokens* de un documento desconocido pueden no haber sido vistos en el conjunto de documentos de entrenamiento y asumir un modelo de carácter uniforme (*character language model*, por sus términos en inglés) para tales *tokens*. La probabilidad de los *tokens* no visto es proporcional al número de caracteres en el *token* e inversamente proporcional al tamaño del conjunto de caracteres.

Cuando las colecciones de aprendizaje son pequeñas, pueden producirse errores al estimar dichas probabilidades. Por ejemplo, cuando un determinado término no aparece nunca en esa colección de aprendizaje, pero aparece en los documentos a categorizar. Esto implica la necesidad de aplicar técnicas de suavizado, a fin de evitar distorsiones en la obtención de las probabilidades. [33]

Modelo Multivariado Bernoulli

En este modelo un documento es representado como un vector binario sobre el espacio de palabras. Es equivalente al modelo binario, el cual genera un indicador para cada término del vocabulario, 1 indica la presencia del término en el documento o 0 la ausencia.[25] De esta manera, el número de ocurrencia no es considerado. Ambos modelos difieren en la representación de los documentos, lo cual influye en la distribución condicional $P(d|c)$, donde la probabilidad de un

documento dada su categoría es simplemente el producto de la probabilidad de los valores de los atributos sobre todos los términos de atributos:

$$P(w_i | c_j) = \prod_{t=1}^T P(w_t | c_j)^{B_{it}} \prod_{t=1}^T (1 - P(w_t | c_j))^{1 - B_{it}}$$

donde $|T|$ es el número de términos en el conjunto de entrenamiento y B_{it} está

definido como: $B_{it} = \begin{cases} 1 & \text{si el término } w \text{ aparece en el documento } d_i \\ 0 & \text{en caso contrario} \end{cases}$

Se puede estimar la probabilidad del término w_t en la categoría c_j como sigue:

$$P(w_t | c_j) = \frac{1 + \sum_{i=1}^N B_{it} \cdot y(w_t | c_j)}{2 + \sum_{i=1}^N y(w_t, c_j)}$$

donde, N es el número de documentos de entrenamiento y $y(w_t, c_j)$ está definida

en la ecuación [78]: $y(w_t, c_j) = \begin{cases} 1 & \text{si } d_i \in c_j \\ 0 & \text{en caso contrario} \end{cases}$

A diferencia del Modelo Multinomial, el Modelo Multivariado de Bernoulli no toma en cuenta el número de veces que ocurre cada término en el documento, e incluye explícitamente la probabilidad de la no ocurrencia de los términos que están ausentes en el documento. [69]

Como se ha dicho anteriormente, la implementación del Naive Bayes es sencilla y rápida y sus resultados son bastante buenos, como atestiguan numerosos trabajos experimentales: [23, 64, 72, 110]; y se aplica de manera exitosa en la clasificación de documentos de texto. [104]

1.4.2.4 TF/IDF

El clasificador TF/IDF usa vectores de rasgos para representar categorías y documentos. Un vector de rasgos es una lista de términos o *tokens* y frecuencias asociados a estos.

Considere una lista de documentos de entrenamiento por múltiples categorías (ver Figura 1.3). La lista de los términos extraídos del texto de los documentos de entrenamiento y las frecuencias, para una categoría, es acumulada en un vector.

El vector creado por cada categoría es normalizado usando el término y el inverso de las frecuencias del documento (IDF).

El TF es normalizado por la raíz cuadrada de los términos de frecuencia:

$$TF = \sqrt{TermFrec}$$

y el IDF de un término x es:

$$IDF(x) = \log\left(\frac{N}{df(x)}\right) \quad (3)$$

donde $df(x)$ es el número de documentos o vectores donde el término x aparece y N es el número de vectores o tamaño de la colección de documentos. El valor del IDF varía entre cero (para un término x que ocurre en todas las categorías) y el $\log(N)$ (cuando un término ocurre exactamente en una categoría).

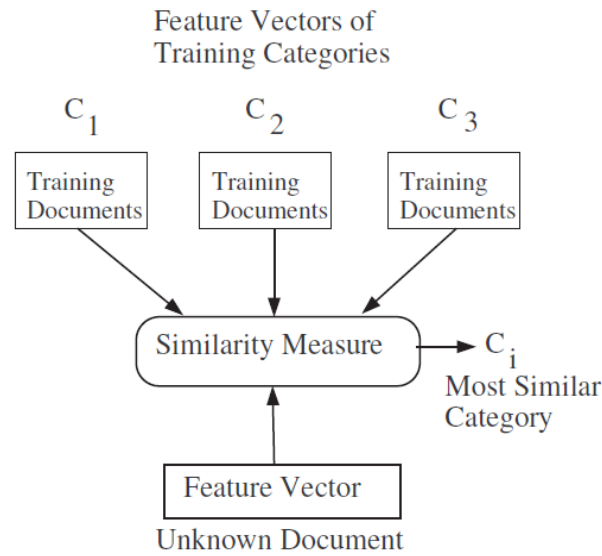


Figura 1.3: Vectores de Categorías de Entrenamiento.

Para evitar los extremos de los valores a obtener con el IDF (cero o $\log(N)$) se modifica la ecuación (3):

$$IDF(x) = \log\left(\frac{N}{df(x) + 1} + 1\right)$$

Quedando finalmente en el vector de rasgos, los términos con el valor de TF/IDF correspondiente a cada uno de ellos:

$$TFIDF = TF * longNorm * IDF$$

La $longNorm = \frac{1}{\sqrt{numTerms}}$, de tal manera que se tenga en cuenta la longitud del

documento, donde $numTerms$ es la cantidad de términos de un documento.

Para clasificar un nuevo documento, se determina la similitud entre los vectores de rasgos por categoría y el documento a clasificar. Para calcular la medida se hace uso de la medida de Manhattan o la suma de las diferencias absolutas entre cada dimensión. El valor mayor arrojado estará asociado con la categoría idónea a clasificar el nuevo documento.

1.4.2.5 K-NN

K-NN es un tipo de aprendizaje basado en memoria, en el cual, cada instancia de entrenamiento representada como un vector de rasgos, es almacenado junto con su categoría.

Para clasificar un documento, se calcula la similitud entre este y los documentos de entrenamiento. La medida utilizada para determinar dicha cercanía es la función de proximidad sobre los vectores de los documentos de entrenamiento y el documento a clasificar:

$$proximidad(u, v_2) = \frac{1}{(+ distancia(u, v_2))}$$

El valor resultante para esta medida se encontrará en el intervalo de 0 a 1 y permitirá encontrar los k ejemplos de entrenamiento más cercanos al objeto que está siendo clasificado. Cada uno de los k vecinos más cercanos vota por su categoría. El resultado de la votación son las puntuaciones (*scores*) de la clasificación y el ejemplo de mayor puntuación será la clase idónea.

Es un algoritmo sencillo y tiene eficacia notable; esto último se puede ver en el caso de los experimentos publicados en [109] sobre diversas colecciones documentales (parte de las cuales estaba constituida por noticias de prensa). Es especialmente eficaz cuando el número de categorías posibles es alto y cuando los documentos son heterogéneos y difusos. [33]

Además, el clasificador K-NN tiene un rendimiento, el cual depende de factores experimentales como las características del conjunto de documento, el número de

ejemplos de entrenamiento por categoría, la medida de similitud, etc. Sin embargo, suele ser difícil encontrar el valor óptimo de k . [96]

1.5 Evaluación de la clasificación

Actualmente, en la literatura se referencia el nacimiento o existencia de un gran número de clasificadores. Muchos de estos, nacen a partir de las extensiones realizadas de los ya existentes o hibridaciones de estos. Las diferencias entre todos ellos son notables. Evidencia de esto, suele ser el desempeño del algoritmo, efectividad o precisión en la clasificación.

Escoger un clasificador a simple vista entre tantos existentes, es una decisión que en determinados momentos puede resultar fácil para un conjunto de datos de baja dimensionalidad, pero casi siempre no es el caso. ¿Qué hacer para uno de alta dimensionalidad?

Por otro lado, la evaluación de los clasificadores no es absoluta, el hecho de que tenga un buen comportamiento para un conjunto de datos no significa que sea de igual manera para otro.

De acuerdo a la literatura actual, existen diversas maneras de realizar dichas evaluaciones con el fin de obtener cuantitativamente las diferencias existentes entre clasificadores bajo determinadas condiciones, las cuales serán tratadas en los subepígrafes [1.5.2](#) y [1.5.4](#).

1.5.1 Corpus textual

Se dispone de un corpus textual de noticias de 725 artículos tomado del periódico *jahora!* digital de la provincia de Holguín. Cada artículo se encuentra preclasificado en una categoría de 8 existentes. (Ver Tabla 1.1).

Categoría	Cantidad de documentos
Deporte	100
Cultura	102
Especial	61
Holguín	102
Nacional	102
Internacional	103
Salud	62
Opinión	93

Tabla 1.1: Categorías y documentos de entrenamiento utilizados.

Con el propósito de escoger el algoritmo con mejor desempeño, de los analizados en [epígrafe 1.4](#), se desarrolló una experimentación. Para cada algoritmo, se efectuó el enfoque validación cruzada (del inglés *k-fold cross-validation*), en el cual k diferentes clasificadores son contruidos particionando el corpus inicial en k conjuntos distintos de aproximadamente igual tamaño. Se realizan k pruebas con $k-1$ partes de entrenamiento y 1 de evaluación, y se promedian los resultados. En este caso, se decidió tomar un valor de $k = 10$ por ser el recomendado en la literatura consultada.

1.5.2 Medidas a utilizar

Las métricas de efectividad más populares en la evaluación de clasificadores para la clasificación de textos son **Precisión** (π , del inglés *Precision*) y **Cobertura** (ρ , del inglés *Recall*)[63], que miden lo correcto y completo del método, respectivamente, y la medida **F**, que es una combinación lineal de ambas. Todas estas medidas son nociones clásicas de Recuperación de Información, adaptadas a la clasificación de textos, donde:

- π : es la probabilidad de si un documento aleatorio d_x es clasificado bajo c_i , esta decisión sea correcta
- ρ : es la probabilidad de que si un documento aleatorio d_x debe ser clasificado bajo c_i , esta decisión sea tomada

Estas probabilidades pueden ser estimadas como se muestra en la Tabla 1.2:

MEDIDA	MICRO-PROMEDIO	MACRO-PROMEDIO
Precisión (π)	$\pi = \frac{\sum_{i=1}^{ C } TP_i}{\sum_{i=1}^{ C } TP_i + FP_i}$	$\pi = \frac{\sum_{i=1}^{ C } \frac{TP_i}{TP_i + FP_i}}{ C }$
Cobertura (ρ)	$\rho = \frac{\sum_{i=1}^{ C } TP_i}{\sum_{i=1}^{ C } TP_i + FN_i}$	$\rho_i = \frac{\sum_{i=1}^{ C } \frac{TP_i}{TP_i + FN_i}}{ C }$

Tabla 1.2: Promedio de Precisión y Cobertura.

donde:

- **Micro-Promedio:** es para dar a las categorías una importancia proporcional al número de ejemplos positivos que le corresponden
- **Macro-Promedio:** todas las categorías importan lo mismo
- las ecuaciones están en términos de la Tabla de Contingencia 1.3 para la categoría c_i

Categoría c_i		Juicio del experto	
		SI	NO
Juicio del clasificador	SI	TP_i	FP_i
	NO	FN_i	TN_i

Tabla 1.3: Matriz de Confusión binaria.

La medida **F** (F_β) puede ser calculada por la función[105]:

$$F_\beta = \frac{\beta^2 + 1 * \pi * \rho}{\beta^2 * \pi + \rho} \quad (4)$$

donde β puede ser vista como el grado de importancia atribuido a π y ρ . Si $\beta=0$ entonces F_β coincide con π , mientras que si $\beta=+\infty$ entonces F_β coincide con ρ . Usualmente, un valor de $\beta=1$ es usado, el cual atribuye igual importancia a π y ρ . La ecuación (4) es considerada como el Micro-Promedio de F y el Macro-Promedio está dado por:

$$F_\beta = \frac{\beta^2 + 1 * \pi * \rho}{\beta^2 * \pi + \rho + |C|}$$

El principal criterio para evaluar clasificadores es la exactitud (del inglés *accuracy*), la cual es la proporción de instancias clasificadas correctamente y el número de instancias en el corpus de prueba. Es decir, a partir de la Matriz de Confusión o Tabla de Contingencia (Tabla 1.3), se divide la suma de la cantidad en la diagonal de la matriz entre el número total de clasificaciones, que es lo mismo que la suma de cada celda en la matriz.

$$Accuracy = \frac{P + TN}{P + TN + FN + FP}$$

Cada una de estas medidas se encuentra implementada en la biblioteca de clases o Interfaz de Aplicación del Programador (API, por sus siglas en inglés, *Application Programmer's Interface*): Lingpipe 3.8.2.

1.5.3 Resultados de la evaluación

A partir del corpus inicial y el enfoque *cross-validation* para 10 folds se entrenaron clasificadores, para luego ser evaluados con las medidas tratadas en el [subepígrafe 1.5.2](#). El propósito de estas evaluaciones es escoger el algoritmo de mejor desempeño en términos de efectividad y tiempo, para ser utilizado en la solución propuesta por la investigación.

Respecto a los valores arrojados del *accuracy*, el clasificador *DynamicLM* se comporta un poco mejor que *LogisticRegression* y por orden descendente le sigue *Tfidf*, *NaiveBayes*, *Knn* y por último *Bernoulli*. (Ver Figura 1.4, Ver [Anexo 2](#))

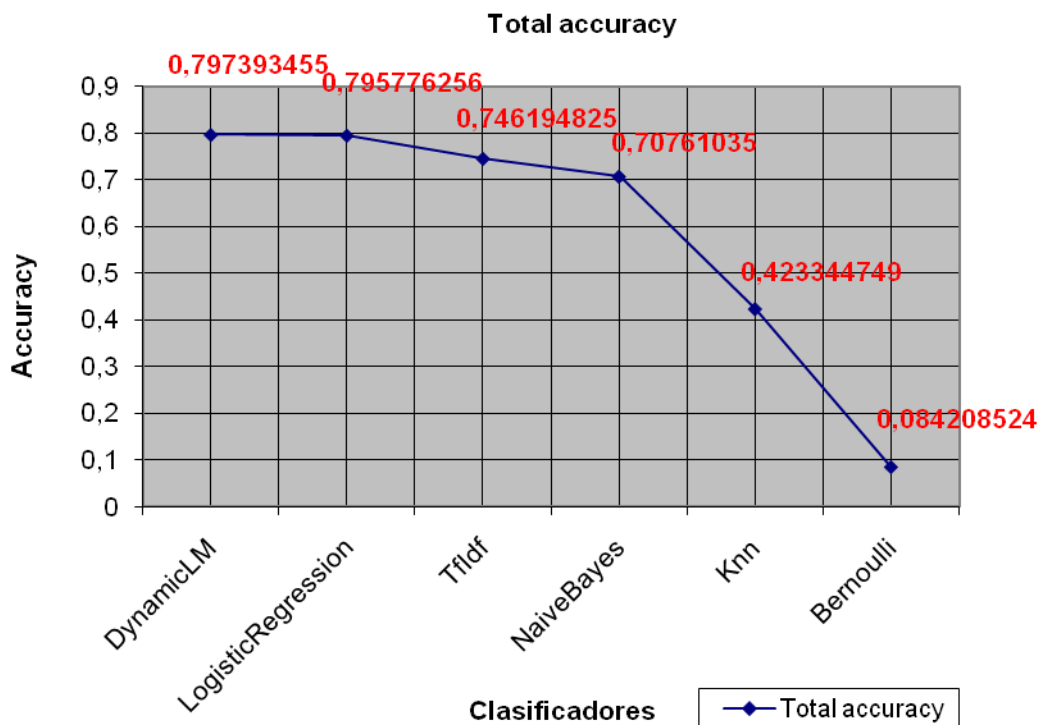


Figura 1.4: Comparación entre los clasificadores teniendo en cuenta el *accuracy*.

En cuanto a las restantes métricas (Macro y Micro Promedio del *Recall*, *Precision* y la medida *F*), el clasificador *DynamicLM* parece comportarse mejor que el resto

de los clasificadores, excepto en el valor de Macro-Promedio *Precision*, que para *LogisticRegression* es un poco mejor. (Ver Figura 1.5, Ver [Anexo 2](#))

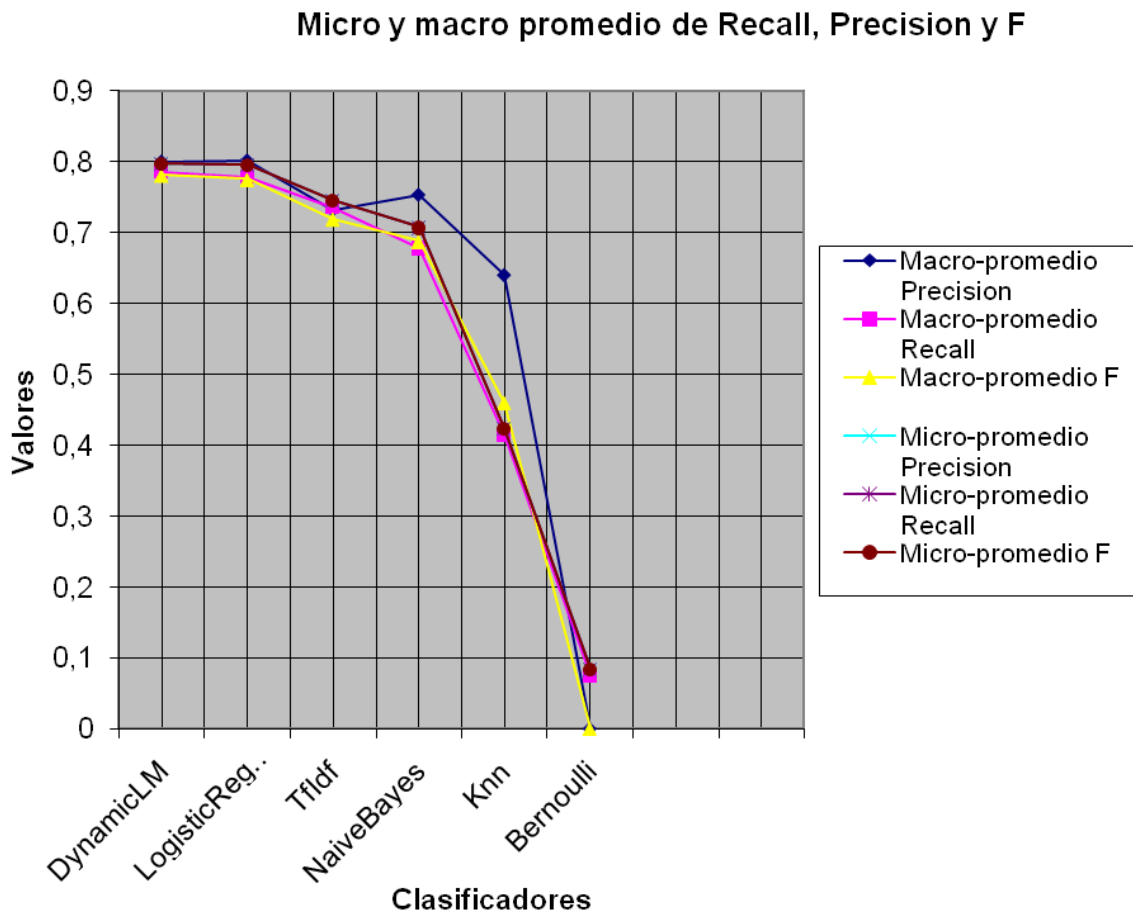


Figura 1.5: Comparación entre clasificadores según las medidas Micro y Macro Promedio de *Recall*, *Precision* y *F*.

De acuerdo al costo en tiempo de ejecución de cada algoritmo (ver Figura 1.6), para una máquina de CPU 2.0GHz y 1.5 GB de RAM, *DynamicLM* y *LogisticRegression*, a pesar de ser los de mejor comportamiento, son los que más se demoran en tiempo de ejecución. Aunque, integrando el aspecto del tiempo a lo anteriormente analizado en las figuras 1.4 y 1.5, se puede apreciar que *DynamicLM*, de manera general, se comporta mejor que *LogisticRegression* para este corpus textual noticioso.

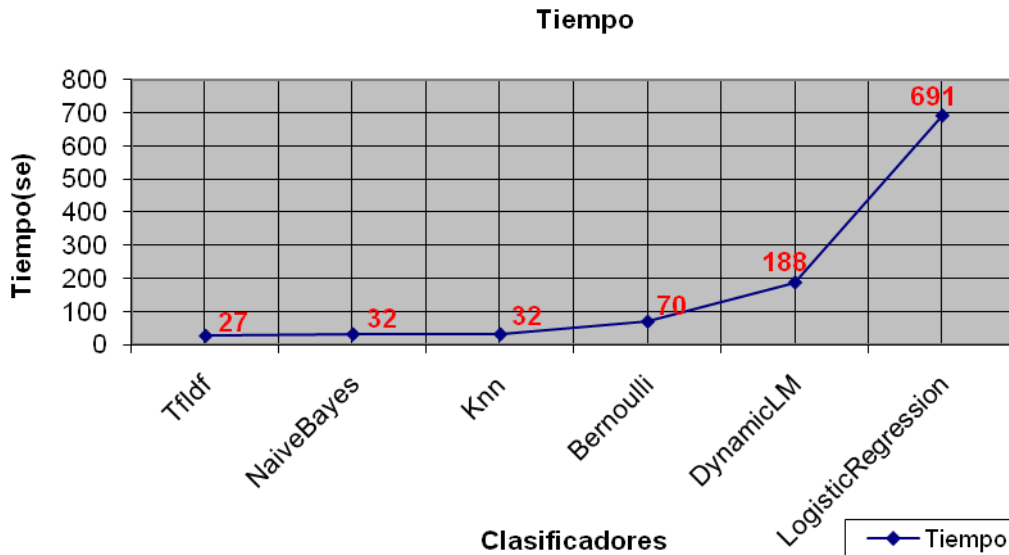


Figura 1.6: Comparación de clasificadores en cuanto al tiempo de ejecución.

1.5.4 Estudio experimental

Actualmente no se han establecido procedimientos para comparar algoritmos sobre múltiples problemas. Los investigadores adoptan diferentes técnicas estadísticas para decidirse si las diferencias entre los algoritmos son verdaderas o arbitrarias. Los análisis estadísticos han sido realizados para encontrar diferencias significativas entre los resultados obtenidos por los métodos estudiados.

Hay principalmente dos grupos de pruebas estadísticas, paramétricas y no paramétricas. En este caso, se utilizaron pruebas no paramétricas tanto para k muestras relacionadas (Prueba de Friedman [37, 38]) como para dos muestras relacionadas (Prueba de Wilcoxon [98]). Cuando se realizaron las pruebas o test de Friedman y de Wilcoxon se aplicó el método de Monte Carlo con un intervalo de confianza de 95 y 99 por ciento, respectivamente, así como un número de muestras igual a 10000 en ambos.

Se considera de manera general para una significación menor que 0.01, diferencias altamente significativa, mientras para una significación menor que 0.05 y mayor que 0.01, diferencias significativas.

Para el análisis estadístico de los resultados, se utilizó la herramienta SPSS (*Statistical Package for Social Sciences*, por sus términos en inglés), versión 15.0

para Windows, ya que es considerado uno de los paquetes de procesamiento estadístico más rigurosos y completos.

Se intenta comprobar la hipótesis nula de que entre los algoritmos (una pareja de ellos o entre todos, según prueba) no hay diferencias significativas; en caso contrario se rechaza la misma.

El primer experimento consiste en probar si existen diferencias entre los algoritmos que se analizan con el Test de Friedman (ver Tabla 1.5), respecto al *accuracy* para 10 *fold* por algoritmo. También se tuvo en cuenta el *ranking* devuelto por los algoritmos según la exactitud de cada uno (ver Tabla 1.4). Como se muestra a continuación sí existen tales diferencias.

Algoritmos de Clasificación	Rango promedio
LogisticRegression	5,30
DynamicLM	5,10
TfIdf	4,30
NaiveBayes	3,30
Bernoulli	1,00
Knn	2,00

Tabla 1.4: Ranking según el *accuracy*.

N				10
Chi-cuadrado				43,295
gl				5
Sig. asintót.				,000
Sig. Monte Carlo	Sig.			,000
	Intervalo de confianza de 95%	Límite inferior		,000
		Límite superior		,000

Tabla 1.5: Test de Friedman.

Luego, para identificar esas diferencias se aplicó un Test de Wilcoxon para dos muestras relacionadas (entre cada uno de los algoritmos). (Ver Tabla 1.6)

Algoritmos	R+	R-	valor p	Hipótesis	Diferencias	Mejor Desempeño
Bernoulli - LogisticRegression	0	55	0,00503351	Rechazada	Altamente sig.	LogisticRegression
Knn - DynamicLM	0	55	0,00503351	Rechazada	Altamente sig.	DynamicLM
Bernoulli - Tfldf	0	55	0,00503351	Rechazada	Altamente sig.	Tfldf
Bernoulli - NaiveBayes	0	55	0,00503351	Rechazada	Altamente sig.	NaiveBayes
Bernoulli - Knn	0	55	0,00503351	Rechazada	Altamente sig.	Knn
NaiveBayes - LogisticRegression	0	55	0,00506203	Rechazada	Altamente sig.	LogisticRegression
Knn - LogisticRegression	0	55	0,00506203	Rechazada	Altamente sig.	LogisticRegression
NaiveBayes - DynamicLM	0	55	0,00506203	Rechazada	Altamente sig.	DynamicLM
Bernoulli - DynamicLM	0	55	0,00506203	Rechazada	Altamente sig.	DynamicLM
Knn - Tfldf	0	55	0,00506203	Rechazada	Altamente sig.	Tfldf
Knn - NaiveBayes	0	55	0,00506203	Rechazada	Altamente sig.	NaiveBayes
NaiveBayes - Tfldf	3	33	0,03524022	Rechazada	Significativas	Tfldf
Tfldf - DynamicLM	8,5	46,5	0,05263321	Aceptada	No hay	DynamicLM
Tfldf - LogisticRegression	5	31	0,06870357	Aceptada	No hay	LogisticRegression
DynamicLM - LogisticRegression	24	31	0,72075492	Aceptada	No hay	DynamicLM

Tabla 1.6: Test de Wilcoxon.

Los algoritmos *Tfldf*, *LogisticRegression* y *DynamicLM* fueron encontrados considerablemente mejores que *NaiveBayes*, *Knn* y *Bernoulli*, el algoritmo *NaiveBayes* resultó considerablemente mejor que *Knn* y *Bernoulli*, y el algoritmo *Knn* fue considerablemente mejor que *Bernoulli*. No hubo diferencias significativas entre los pares *Tfldf* – *DynamicLM*, *Tfldf* – *LogisticRegression* y *DynamicLM* – *LogisticRegression*, aunque resultó mejor el *DynamicLM* en dos ocasiones.

A partir de los análisis expuestos anteriormente y debido a que en la mayoría de los casos resultó mejor *DynamicLM*, se decidió utilizar este algoritmo para el desarrollo de la aplicación.

1.6 Descripción de las tecnologías para la construcción de la solución propuesta

Con el desarrollo de las TIC la sociedad está altamente interconectada, en la llamada era de la información, donde el software es cada vez más el gran intermediario entre la información y la inteligencia humana, a raíz de la progresiva informatización de casi la totalidad de las empresas. Por esta razón, es muy importante la libertad para poder acceder a la información, así como, conocer

quién controla el software y qué garantías tenemos de su transparencia y fiabilidad.

El movimiento del Software Libre tuvo su origen en el mundo académico. Desde hace más de treinta años, numerosos programadores de distintas universidades han desarrollado herramientas de forma cooperativa y abierta, intercambiando libremente su código fuente.[99]

El software libre es aquel que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las 4 libertades que lo caracterizan[80]:

- Libertad 0: La libertad de usar el programa, con cualquier propósito.
- Libertad 1: La libertad de estudiar cómo funciona el programa y poder adaptarlo a las necesidades de cada cual gracias a la disponibilidad de acceso al código fuente y a que la licencia lo permite.
- Libertad 2: La libertad de redistribuir copias.
- Libertad 3: La libertad de mejorar un programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. El acceso al código fuente es un requisito previo para esto.

Dentro de software libre hay matices que es necesario tener en cuenta. Por ejemplo, el software de *dominio público*⁸ significa que no está protegido por el Copyright⁹, por lo tanto, podrían generarse versiones no libres del mismo; en cambio el software libre protegido con Copyleft¹⁰ impide a los redistribuidores incluir algún tipo de restricción a las libertades propias del software así concebido, es decir, garantiza que las modificaciones seguirán siendo software libre. También es conveniente no confundir el software libre con el software gratuito; éste no cuesta nada, hecho que no lo convierte en software libre, porque no es una cuestión de precio, sino de libertad.[80]

⁸ Término legal que significa de manera precisa “sin copyright”.

⁹ Copyright: Término usado para denominar al software que no es libre ni de fuente abierta.

¹⁰ Copyleft: Es un caso particular de software libre cuya licencia obliga a que las modificaciones que se distribuyan sean también libres.

Una de las categorías de software libre es código abierto (del inglés *open source*); se centra en el potencial de realización de software de alta calidad, pero esquiva las ideas de libertad, comunidad y principio. En 1998, una parte de la comunidad decidió dejar de usar el término “*free software*” (software libre) y usar “*open source software*” (software de código abierto), con el propósito de evitar la confusión de “*free*” con “gratis”. Otros, sin embargo, apuntaban a apartar el espíritu de principios que ha motivado el movimiento del software libre, y resultar así atractivos a los ejecutivos y usuarios comerciales.[80]

La migración a software libre se incrementa en países que como Cuba, sufren de grandes limitaciones para acceder a cualquier software propietario, y que resulta extremadamente costoso por realizarse la adquisición mediante intermediarios. Para Cuba, la migración hacia el software libre constituye una necesidad táctica y estratégica. En primer lugar por la soberanía tecnológica que nos proporciona y la seguridad en términos del soporte informático. En segundo lugar fortalece la invulnerabilidad económica, política y militar del país, al eliminar un posible pretexto de invasión extranjera por el uso no autorizado de software propietario. Y en tercer lugar nuestro país posee todas las condiciones para llegar a convertirse en una potencia mundial en el desarrollo del software libre.[15]

1.6.1 Herramientas para preprocesar y clasificar documentos

Se analizaron algunos Sistemas de Recuperación de Información (SRI) y herramientas de Procesamiento del Lenguaje Natural, que permitieran realizar tareas de procesamiento y clasificación de texto, y que a su vez fueran libres y de código abierto. Algunas de ellas son: Lemur, Indri, Weka, Lucene, LingPipe. A continuación se realiza un estudio comparativo de éstas, destacando sus principales características, ventajas y desventajas. En el [Anexo 3](#) se muestra una tabla con los resultados de la comparación de las mismas.

1.6.1.1 Lemur

Lemur es una aplicación *framework*¹¹ libre de código abierto diseñada para facilitar la investigación de modelado de lenguaje¹² y RI, así como la construcción de software relacionados con esos temas. En la recuperación usa modelado de lenguaje, para lo que utiliza Indri y divergencia-KL¹³, así como también el Modelo Espacio-Vectorial, TF/IDF, Okapi e Inquiry¹⁴. [59]

Esta herramienta forma parte del Proyecto Lemur (<http://www.lemurproject.org>), que es una colaboración entre el Centro para Recuperación de Información Inteligente del Departamento de Ciencia de la Computación en la Universidad de Massachusetts y el Instituto de Tecnologías de Lenguaje de la Escuela de Ciencia de la Computación en la Universidad Carnegie Mellon, y que comprende además, el motor de búsqueda Indri. [59]

El sistema está escrito en C y C++ y proporciona interfaces interactivas para Windows, Linux y la Web. Dispone de APIs Java, C++ y C#. [61]

Permite documentos TREC, Web TREC, texto plano, HTML, XML, PDF, Microsoft Word y Microsoft PowerPoint. [59]

Esta herramienta incluye tecnologías de recuperación ad-hoc¹⁵ y distribuida, RI de lenguaje cruzado, sumarización, filtrado, agrupamiento y clasificación. [60] Realiza

¹¹ Una aplicación o *software framework* es una abstracción en la cual el código común proporcionando funcionalidad genérica puede ser selectivamente sobrescrito o especializado por el código del usuario, proporcionando funcionalidad específica. Los *frameworks* o plataformas son un caso especial de bibliotecas de *software* o APIs.

¹² *Framework* que representa típicamente un documento como una secuencia de *tokens*. Un modelo de lenguaje estadístico asigna una probabilidad a una secuencia de *tokens*.

¹³ En teoría de la probabilidad la divergencia de Kullback-Leibler es un indicador de la similitud entre dos funciones de distribución. Dentro de la teoría de la información también se la conoce como divergencia de la información, ganancia de la información o entropía relativa.

¹⁴ Motor de búsqueda basado en el *framework* de red de inferencia que soporta consultas estructuradas y usa probabilidad basada en TF/IDF.

¹⁵ La recuperación ad-hoc es la tarea estándar de RI de encontrar documentos que son temáticamente relevantes para una necesidad dada de información (consulta) en una máquina.

tokenización del texto y emplea los métodos Porter y Krovetz para la lematización y segmentación.

1.6.1.2 Indri

Indri es un SRI de código abierto del proyecto Lemur. Combina el modelado de lenguaje y técnicas de red de inferencia¹⁶ para la RI. [70]

Se puede usar también como biblioteca para construir otras aplicaciones desde Java, C++, C# y PHP. Procesa documentos de texto, HTML, XML, PDF y documentos Word y PowerPoint en Windows. Puede analizar documentos TREC, cables de noticias TREC y colecciones Web TREC, y retornar los resultados en el formato estándar TREC.[62]

Permite la tokenización independiente del lenguaje de documentos codificados en UTF-8, la lematización y el *stemming*. [62]

Algunas de las características de ejecución de Indri son[100]:

- Estructuras de memoria: Mantiene datos en la memoria tanto tiempo como sea posible para evitar accesos al disco.
- Aislamiento del cerrojo: El sistema mantiene cerrojos exclusivos a datos tan poco tiempo como sea posible.
- Estructuras de solo lectura: El sistema usa estructuras de datos de sólo lectura cuando sea posible para reducir la necesidad de cerrojos exclusivos.
- Entrada/Salida en segundo plano: El sistema interactúa con el subsistema de disco de alto costo en hilos de fondo, así las operaciones de procesamiento pueden ser de bajo costo.
- Estructuras multiversión: Cuando las operaciones largas deben acceder a datos que cambian, permite la existencia de múltiples versiones de datos simultáneamente para reducir la necesidad de cerrojos exclusivos.

¹⁶ El modelo de red de inferencia proporciona una forma de combinar muchas fuentes de evidencias de relevancia de documento. En este modelo una consulta está compuesta por una serie de conceptos que pueden ser términos, frases o entidades más complejas. Un documento es relevante para un usuario cuando contiene los conceptos listados en la consulta.

1.6.1.3 Weka

Weka (*Waikato Environment for Knowledge Analysis*, por sus términos en inglés, Entorno para Análisis del Conocimiento de la Universidad de Waikato, <http://www.cs.waikato.ac.nz/ml/weka/>) es un software para aprendizaje automático y minería de datos, un conjunto de librerías Java para la extracción de conocimientos desde BD. Es libre, de código abierto y ha sido desarrollado en la Universidad de Waikato, Nueva Zelanda, bajo la licencia GNU-GPL, lo cual ha posibilitado que sea una de las herramientas más utilizadas en el área en los últimos años, en particular con finalidades docentes y de investigación.[4, 32]

Posee una Interfaz Gráfica de Usuario (GUI, por sus siglas en inglés, *Graphics User Interface*) muy fácil de usar y una colección de herramientas de visualización y algoritmos de aprendizaje, que pueden ser aplicados directamente a una colección de datos o llamados desde una aplicación Java.

Contiene herramientas para preprocesamiento de datos, clasificación, regresión, agrupamiento, reglas de asociación, visualización y selección. Es también utilizado para desarrollar nuevos esquemas de aprendizaje automático y permite fácilmente añadir extensiones y modificar métodos.[32]

1.6.1.4 Lucene

Jakarta Lucene (<http://jakarta.apache.org/lucene/>) es un API de alto rendimiento para la RI desarrollado en Java, escalable, multiplataforma y de código abierto. Fue escrito por Doug Cutting¹⁷. Está apoyado por *Apache Software Foundation* y se distribuye bajo la *Apache Software License*. Tiene versiones para otros lenguajes y puede ser fácilmente integrado en muchos tipos de aplicaciones y extendido para acomodarse a necesidades especiales.[97]

Lucene parsea los documentos para convertir el *stream* en un formato de *tokens* de texto plano para que los pueda digerir y así procesar el contenido. Utiliza clases analizadoras para tokenizar el texto, eliminar las *stop words*, convertir las

¹⁷ Lucene es el segundo nombre de la esposa de Doug y el primer nombre de su abuela materna.

mayúsculas en minúsculas, aplicar técnicas de *stemming* para el idioma Inglés, chequear la homogeneidad ortográfica (*spelling*), entre otras funciones. [87]

Lucene puede procesar texto desde muchas fuentes: XML, PDF, HTML, TXT, documentos Word entre los más conocidos, lo que hace a Lucene muy usado por muchas aplicaciones reales. [97]

1.6.1.5 LingPipe

LingPipe (<http://alias-i.com/lingpipe>) es un API o biblioteca de clases Java desarrollado por *Alias-I, Incorporated* para el análisis lingüístico de Lenguaje Humano o Natural. [13]

Es una suite de avanzada tecnología que realiza tokenización, eliminación de palabras vacías, *stemming*, detección de frases, resolución de co-referencias, clasificación, agrupamiento, acoplamiento difuso de diccionarios, etc. [13]

Es neutral respecto a los objetos de entrada, procesa texto plano, en formato HTML, XML y otros. Procesa documentos en casi cualquier lenguaje humano, incluso algunos que no lo son como el Klingon. Puede manejar innumerables documentos simultáneamente, solo limitado por el rendimiento del procesador o CPU. [13]

Algunas de las herramientas que ofrece LingPipe para la extracción de información y minería de datos son [13]:

- Rastreo de menciones de entidades
- Asociación de menciones de entidad a las entradas de una BD
- Descubrimiento de relaciones entre entidades y acciones
- Correcta ortografía respecto a una colección de textos
- Etiquetamiento de categorías gramaticales y división de frases
- Agrupamiento de documentos por el tema implícito y descubrimiento de tendencias significativas con el paso del tiempo
- Clasificación de texto por lenguaje, carácter de la codificación (*character encoding*), género, tema o sentimiento

La arquitectura de LingPipe está diseñada para ser eficiente, dimensionable, reutilizable y robusta. Algunos puntos de interés incluyen [13]:

- Multilingüe, multi-dominio, modelos multi-género
- N-mejor salida con estimaciones estadísticas de confianza
- Modelos seguros de hilos y decodificadores para sincronización de lectura corriente y escritura exclusiva (*concurrent-read exclusive-write*)
- Carácter sensitivo de la codificación (*character encoding-sensitive*) en la entrada y salida.

Para el desarrollo del software propuesto como solución se escogió Lucene para el procesamiento de los documentos y LingPipe para la clasificación de los mismos, debido a que realizan las tareas necesarias para el software propuesto, son muy fáciles de usar y existe suficiente información sobre ellos.

1.6.2 Herramientas de desarrollo

Para el desarrollo del sistema propuesto como solución se utilizó un Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés, *Integrated Development Environment*). Un IDE no es más que un programa informático compuesto por un conjunto de herramientas de programación, que puede dedicarse en exclusiva a un sólo lenguaje de programación o bien permitir utilizar varios.

1.6.2.1 Eclipse

Eclipse es un IDE de código abierto multiplataforma para desarrollar lo que el proyecto llama "aplicaciones de cliente enriquecido", opuesto a las aplicaciones "cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar IDEs. El proyecto Eclipse define su software como: "*una especie de herramienta universal, un IDE abierto y extensible para todo y nada en particular*".[83]

La plataforma de cliente enriquecido (RCP, por sus siglas en inglés, *Rich Client Platform*) está compuesta por los siguientes componentes[83]:

- Plataforma principal - inicio de Eclipse, ejecución de *plug-ins*
- OSGi - una plataforma para establecer relaciones estándar

- El *Standard Widget Toolkit* (SWT, por sus siglas) - Una herramienta *widget*¹⁸ portable
- *JFace* - manejo de archivos, manejo de texto, editores de texto
- El *Workbench* de Eclipse - vistas, editores, perspectivas, asistentes

La interfaz de usuario de Eclipse también tiene una capa GUI intermedia llamada *JFace*, la cual simplifica la construcción de aplicaciones basada en SWT.[83]

El IDE de Eclipse emplea módulos para proporcionar toda su funcionalidad a la plataforma, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos hace a la plataforma ligera para componentes de software.[83]

1.6.2.2 NetBeans

NetBeans es un IDE libre y gratuito sin restricciones de uso que permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software o módulos. Un módulo es un archivo Java que contiene clases de Java para interactuar con las APIs de NetBeans y un archivo especial (manifest.mf) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos.[93]

Facilita mejoras de lenguaje y plataforma para que los desarrolladores puedan crear aplicaciones empresariales más fácilmente y con menos código, acelerando el tiempo de desarrollo de aplicaciones. [93]

Permite la creación de grandes aplicaciones de escritorio y ofrece servicios comunes para este tipo de aplicaciones, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre sus características están:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas)

¹⁸ En el contexto de la programación de aplicaciones visuales, los *widgets* tienen un amplio significado como componente gráfico o control visual que el programador reutiliza, con el cual el usuario interactúa. Puede ser la ventana de una interfaz gráfica. Suelen reunirse varios *widgets* en juegos de herramientas de *widgets* que son usadas por los programadores para construir interfaces gráficas de usuario.

- Administración de las configuraciones del usuario
- Administración del almacenamiento (guardar y cargar cualquier tipo de dato)
- Administración de ventanas
- *Framework* basado en asistentes (diálogos paso a paso)

Este IDE fue el escogido por las facilidades que brinda en cuanto a la ayuda y al diseño e implementación de aplicaciones de escritorio.

1.7 Fundamentación de la metodología de desarrollo utilizada

En el proceso de desarrollo de software intervienen innumerables variables de las más diversas naturalezas, algunas con comportamiento sumamente difuso o imprevisible, que minan constantemente el avance hacia el éxito y hacen de sus rutinas y decisiones tareas altamente riesgosas, difíciles para controlar la calidad y eficiencia y cuantificar su eficacia.

Desde hace bastante tiempo existe una alternativa: el uso de una metodología, que no es más que un conjunto de procedimientos para la realización de un nuevo software[8], que impone un proceso disciplinado con el fin de hacerlo más predecible y eficiente.[35] Consiste en un lenguaje de modelamiento y un proceso. El lenguaje de modelamiento es la notación gráfica, que incluye diferentes tipos de diagramas. El proceso define quién debe hacer qué, cuándo y cómo alcanzar un objetivo. [79]

Actualmente existen metodologías que permiten desarrollar software de superior calidad, debido a la facilidad de control que éstas proporcionan, aparejado a la posibilidad de concebir inicialmente las bases para el desarrollo del software y su éxito en el tiempo y costo fijados. Cada metodología, a pesar de suministrar los aspectos antes mencionados, posee características peculiares que facilitan la selección de una de éstas a partir de las necesidades de la organización y las características específicas del proyecto a desarrollar.

En los últimos tiempos han cobrado auge las metodologías ágiles, debido a que cada vez más los desarrolladores necesitan obtener aplicaciones en menor

tiempo, más vistosas y de menor costo; y los usuarios exigen calidad, sistemas fáciles de mantener, extender y modificar.[79]

El objetivo principal de las metodologías ágiles es minimizar la documentación de desarrollo, considerando el código fuente la parte más importante y el software que funciona como la principal medida del progreso. Considera a los usuarios finales como parte del equipo de desarrollo, por la importancia que concede al trabajo conjunto de ambos durante todo el proceso.[10]

Surgen como reacción a las metodologías monumentales o pesadas, ya que éstas últimas son muy burocráticas; hay tanto que hacer para seguir la metodología que el ritmo entero del desarrollo se retarda.[35] Son adaptables en lugar de predictivas, por lo que aceptan el cambio como bienvenido, se adaptan y crecen en el cambio, incluso al punto de cambiarse ellas mismas. Son orientadas a la gente y no orientadas al proceso, por lo afirman que el papel del proceso es apoyar al equipo de desarrollo en su trabajo y no al contrario.[35]

Dentro de estas metodologías, algunas de las más nombradas en la literatura son:

- XP (*eXtreme Programming*, por sus términos en inglés), con gran énfasis en las pruebas
- Scrum, que se enfoca principalmente en la planeación iterativa y el seguimiento del proceso, generalmente para proyectos de equipos
- DSDM (Método de Desarrollo de Sistema Dinámico, por sus términos en español), que empieza con un estudio de viabilidad para conocer si DSDM es apropiado para el proyecto
- FDD (Desarrollo Manejado por Rasgos, por sus términos en español), que define 2 tipos de desarrolladores: dueños de clases y programadores jefe
- ICONIX, relativamente ágil y lo suficientemente robusta para un proyecto de mediana envergadura

Teniendo en cuenta lo anteriormente planteado, la cantidad de bibliografía disponible y que el proyecto es pequeño y cuenta con un solo desarrollador, se determinó escoger para el desarrollo del proyecto la metodología ICONIX, proceso simplificado (comparado con otros más tradicionales) de desarrollo de software, orientado a objeto que emplea el Lenguaje Unificado de Modelado (UML, por sus

siglas en inglés, *Unified Model Language*). Está entre la complejidad de RUP (*Rational Unified Process*, por sus términos en inglés), que es una de las metodologías tradicionales y la simplicidad de XP, sin descartar las etapas de análisis y diseño; siendo el primero muy útil para software industriales y el segundo muy útil para software pequeños; por tanto, ICONIX es una mezcla entre la agilidad de XP y la robustez de RUP.

1.7.1 ICONIX

El ciclo de vida de un proyecto, según ICONIX, se divide en cuatro fases principales: Definición de Requerimientos; Análisis, Diseño Conceptual y Arquitectura Técnica; Diseño detallado e Implementación; y Prueba, (Figura 1.7) la cual se puede añadir a conveniencia del desarrollador y teniendo en cuenta las características propias del sistema.

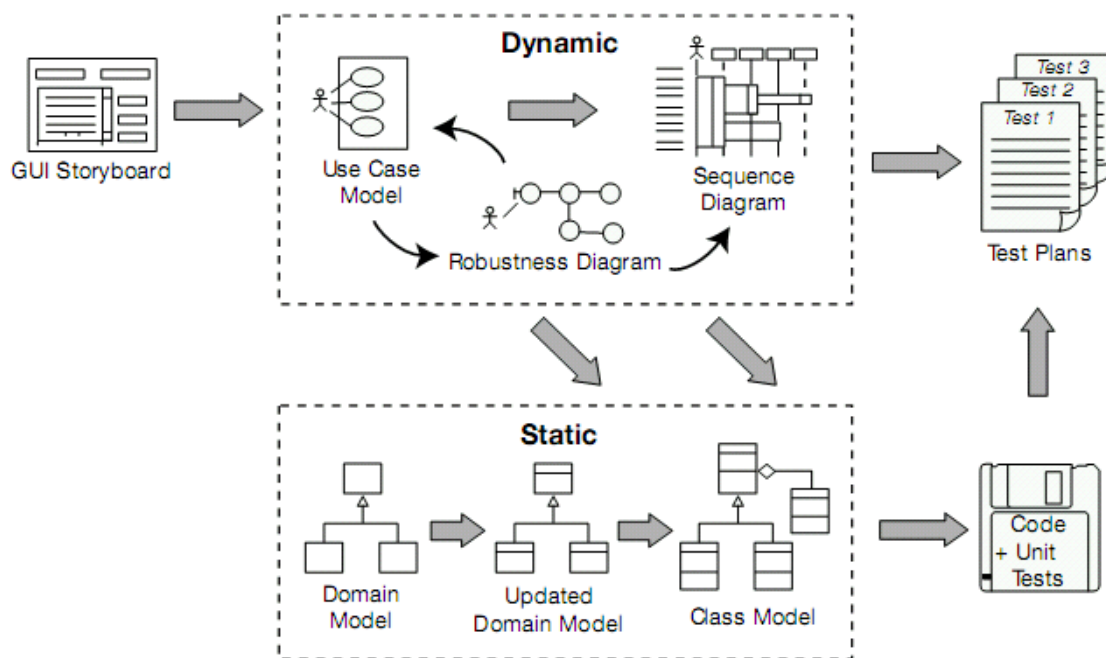


Figura 1.7: Fases de ICONIX.

El proceso de ICONIX es un modelado de objetos conducido por casos de uso, como RUP; también es relativamente pequeño y firme, como XP, pero no desecha el análisis y diseño de éste. Este proceso también hace uso dinámico del UML, ya que se pueden usar solo algunos diagramas, sin exigir la utilización de todos como

en el caso de RUP; aun así es posible seleccionar otros aspectos del UML para complementar los materiales básicos. Esto brinda un enfoque flexible y abierto mientras que guarda un enfoque afilado en el seguimiento de requisitos.

Otras de las características más importantes son: el enfoque iterativo e incremental, ya que se pueden refinar las distintas fases a medida que se vayan identificando nuevos objetos; y la trazabilidad, permitiendo seguir cada caso de uso durante todo el proceso. Además, es centrado en datos, es decir, se descompone en fronteras de datos y es basado en escenarios que descomponen los casos de uso.

Las características antes mencionadas garantizan que la metodología ICONIX sea la apropiada para llevar a cabo el proyecto que se emprende.

Conclusiones del capítulo

En este capítulo se explicó el proceso de edición del periódico *jahora!* digital que se lleva a cabo en la Casa Editora de Holguín. A partir de esto se definieron los conceptos principales asociados al problema y al objeto de estudio, y se decidió utilizar el algoritmo de clasificación *DynamicLM* en el producto de software requerido, ya que es el de mejor desempeño para el campo de acción definido.

Por otra parte, se resaltaron las características, ventajas y desventajas de las tecnologías de desarrollo para una adecuada elección y combinación de éstas, eligiéndose Lucene y LingPipe para el desarrollo de la solución propuesta.

Además, se decidió el empleo de la metodología ICONIX para el desarrollo del sistema por las facilidades que ésta brinda.

Capítulo 2: Descripción y construcción de la solución propuesta

En este capítulo se especifican los requerimientos o requisitos funcionales y no funcionales que debe cumplir la aplicación. Se describe el funcionamiento interno del proceso editorial del diario Web *jahora!* de la Casa Editora a través del Modelo de Dominio, así como el Modelo de Casos de Uso con los Diagramas de Robustez y de Secuencia correspondientes, mediante la metodología ICONIX y el lenguaje UML.

Se realiza una valoración de sostenibilidad del sistema propuesto, de acuerdo con las dimensiones administrativa, socio-humanista, ambiental y tecnológica.

2.1 Planificación

Para el desarrollo de la aplicación es imprescindible el uso de la Ingeniería de Software, ya posibilita un acercamiento sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software, unido a la utilización de métodos y procedimientos de la Gestión de Software, para organizar cada tarea incluyendo las fases que establece la metodología. Esta rama de la ingeniería es la encargada de establecer los principios necesarios para la obtención de un software económico, fiable y eficiente.[46]

Un proyecto es un elemento organizativo a través del cual se gestiona el desarrollo del software. Como el proyecto es realizado por una sola persona, se utiliza para su organización el Proceso Software Personal (PSP, por sus siglas en inglés), que es un marco de trabajo que muestra cómo estimar y planificar el trabajo, cómo controlar el rendimiento frente a esos planes y cómo mejorar la calidad de los programas que se producen.[46]

Para estimar y planificar el trabajo, se identificaron las distintas tareas que lo componen y se analizaron detalladamente, lo que permitió evaluar el tamaño de cada una, determinar el tiempo probable de duración y establecer planes razonables para desarrollarlas, teniendo en cuenta los posibles cambios del proyecto de software. Esto se registra haciendo uso del diagrama de Gantt, para mostrar cuándo empieza cada tarea y cuándo termina. (Ver [Anexo 4](#))

2.2 Definición de Requerimientos

La definición de requerimientos es un proceso fundamental para el éxito de un proyecto de desarrollo. Comprende la determinación de las necesidades o condiciones que debe satisfacer el producto propuesto como solución.

2.2.1 Requerimientos funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar sin tomar en consideración ningún tipo de restricción física.[47] Especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A continuación se muestra la lista con los requerimientos que debe cumplir la solución propuesta.

Lista de Requerimientos Funcionales

El sistema deberá ser capaz de:

1. Clasificar automáticamente las noticias internas desarrolladas por los periodistas en las categorías definidas en el periódico ¡ahora! digital, para cada publicación
2. Cargar las noticias seleccionadas por el Editor Web, que se encuentran almacenadas en el disco duro de la computadora
3. Clasificar las noticias escogidas por el Editor Web en las categorías definidas
4. Visualizar el contenido de las noticias
5. Guardar los resultados de la clasificación

2.2.2 Modelo del Dominio

El Modelo del Dominio es una parte esencial del proceso de ICONIX. Construye un modelo estático inicial del dominio del problema, que será refinado y actualizado durante las posteriores fases del proceso de desarrollo, por lo que siempre reflejará el entendimiento actual del problema. [89]

El término "dominio" se refiere al área que abarca conceptos del mundo real relacionados con el problema que el sistema pretende resolver.[3] El Modelo del Dominio es un modelo conceptual donde se describen los objetos (las clases) del

sistema y sus relaciones. Estos términos (sustantivos y frases sustantivas) constituirán el vocabulario del sistema y muchos de ellos son extraídos de los requerimientos funcionales.[11]

A continuación se muestran los objetos del dominio y el diagrama del modelo (ver Figura 2.1). Estos objetos son algunos de los conceptos fundamentales extraídos de los requerimientos funcionales.

Objetos del dominio:

- noticias internas: noticias elaboradas por los periodistas del periódico *¡ahora!* digital almacenadas en la BD
- periodistas: personas encargadas de elaborar las noticias internas y almacenarlas en el Sitio Web
- categorías: categorías definidas por el periódico *¡ahora!* digital, en la que son clasificados los trabajos periodísticos
- periódico ¡ahora! digital: Web del Periódico *¡ahora!*, de Holguín
- publicación: cada actualización del Sitio Web
- Editor Web: encargado de clasificar las noticias enviadas por los periodistas y las extraídas, por él mismo, de fuentes cubanas externas al periódico *¡ahora!* digital
- noticias: noticias almacenadas en el disco duro de la computadora

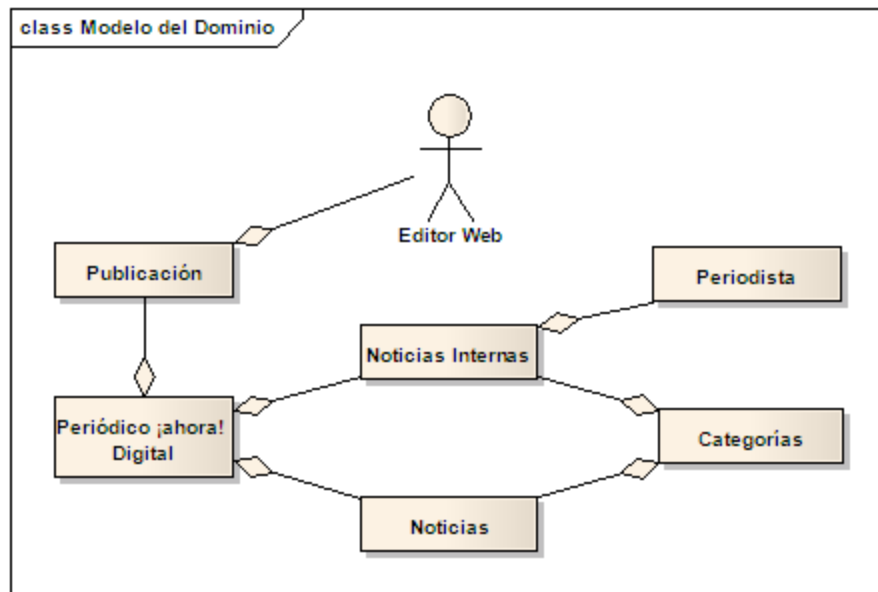


Figura 2.1: Diagrama del Modelo del Dominio.

2.2.3 Modelo de Casos de Uso

A partir de los requerimientos funcionales capturados se desarrolla el Modelo de Casos de Uso, que comprende la identificación de los actores y casos de uso del sistema. No obstante, un importante aspecto de ICONIX es que un requisito se distingue explícitamente de un caso de uso; este último describe un comportamiento mientras que un requisito describe una regla para el comportamiento. Los actores representan entidades externas al sistema, personas o sistemas, y son análogos a un rol del usuario. [89]

ICONIX asume que el Modelo del Dominio inicial es incorrecto y provee un mejoramiento incremental del mismo a medida que se analizan los casos de uso. [89]

Para una mejor comprensión del modelo, los casos de uso se agrupan en subsistemas o paquetes, según las relaciones existentes entre ellos, como se muestra en la Figura 2.2.

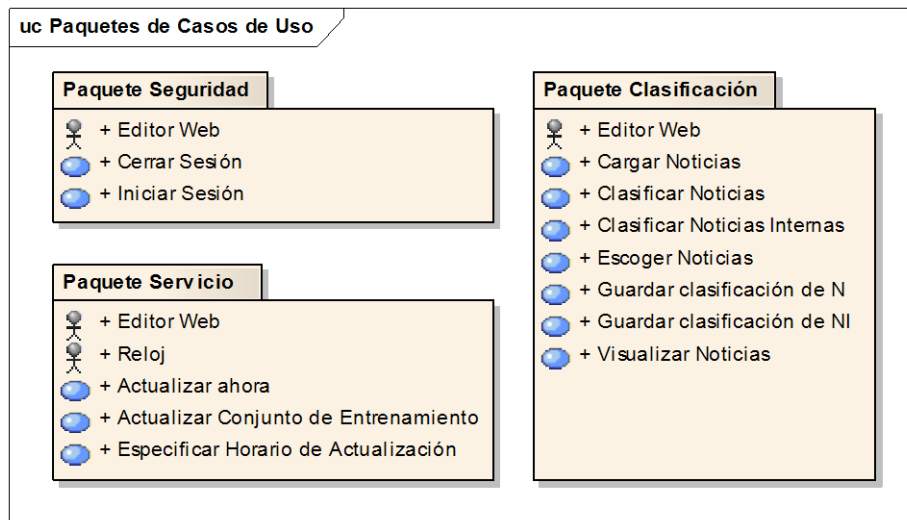


Figura 2.2: Diagrama de Paquetes de Casos de Uso.

Para cada subsistema o paquete se realiza un diagrama de casos de uso (ver Figuras 2.3 – 2.5), relacionando los actores del sistema con los casos de uso correspondientes.

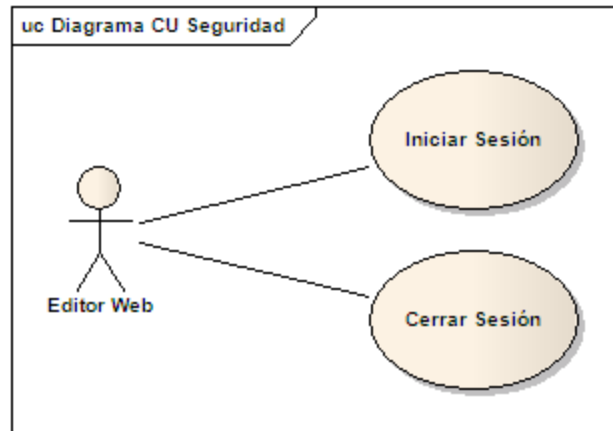


Figura 2.3: Diagrama de Casos de Uso Paquete Seguridad.

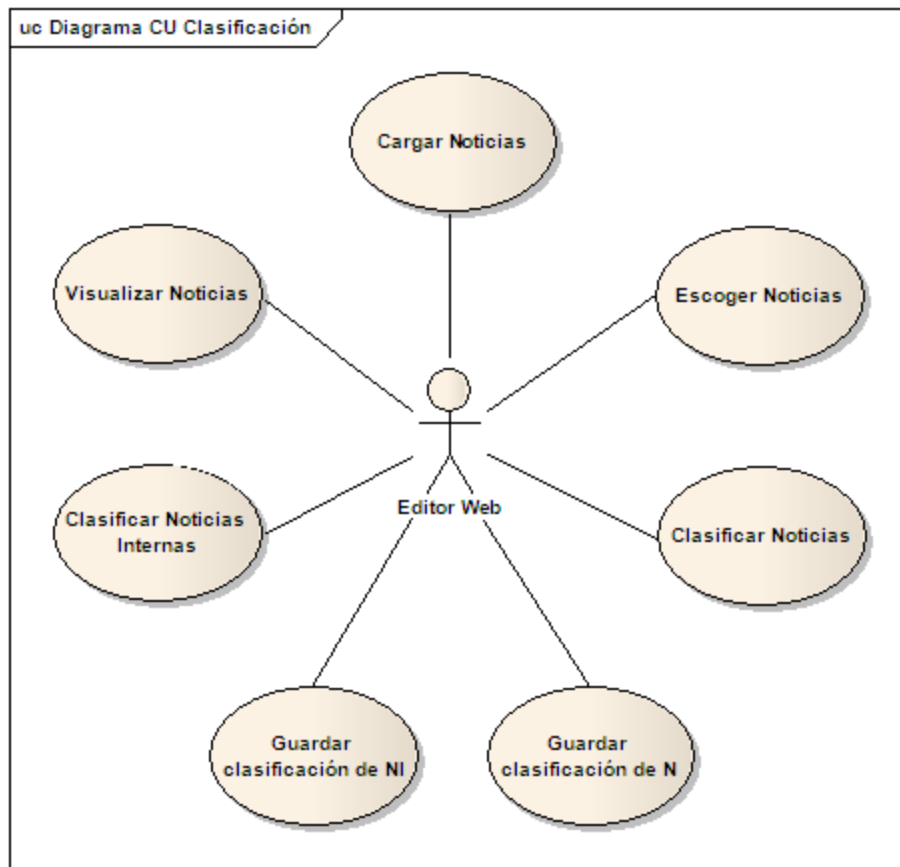


Figura 2.4: Diagrama de Casos de Uso Paquete Clasificación.

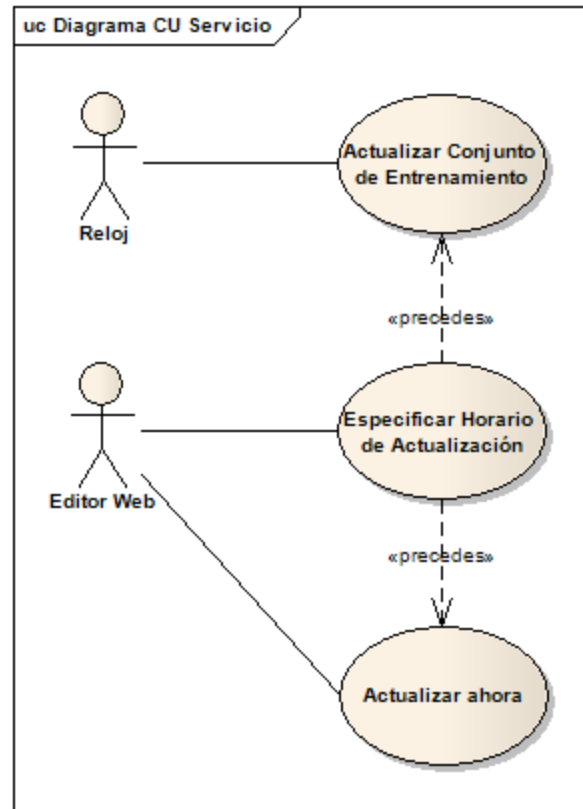


Figura 2.5: Diagrama de Casos de Uso Paquete Automatización.

2.2.3.1 Descripción de Casos de Uso

La descripción de cada caso de uso facilita especificar la secuencia de acciones que el sistema debe llevar a cabo al interactuar con los actores, incluyendo cursos alternativos dentro de la secuencia. Esta tarea, según la metodología ICONIX, se explica siguiendo la regla de los dos párrafos, utilizando voz activa, reflejando la acción del actor y la respuesta del sistema y empleando una estructura de oración sustantivo-verbo-sustantivo.[90] Se especifican, además, los cursos básicos y alternos, en caso de existir. (Ver [Anexo 5](#))

2.2.4 Requerimientos No Funcionales

Una de las tareas opcionales de esta etapa de la metodología que tiene gran influencia en el éxito del producto y en la satisfacción del cliente es la captura de los requerimientos no funcionales. Estos, a pesar de no alterar la funcionalidad del sistema, son muy importantes porque definen las cualidades o propiedades que el

sistema debe poseer. Se deben ajustar a la arquitectura técnica del sistema y se precisan con el objetivo de lograr la aceptación de los usuarios finales, así como el buen funcionamiento, la flexibilidad y la escalabilidad.[47]

Lista de Requerimientos No Funcionales

Apariencia o Interfaz externa

- El sistema debe ser fácil de usar, pues los usuarios finales no son expertos en computación, a pesar de hacer uso de la computadora para realizar las labores rutinarias.
- El diseño debe ser agradable y atractivo a los usuarios para lograr una mejor concentración, sin desviar demasiado su atención del contenido de trabajo.
- Se utilizarán los colores en la gama del negro, rojo, gris y blanco para mayor compatibilidad con los colores del periódico. Las tonalidades serán suaves y relajantes, para evitar esfuerzo visual.
- El idioma a utilizar será el español.
- El sistema debe ser interactivo, por lo que de no poderse ejecutar una acción se visualizará un mensaje de error.

Usabilidad

- El diseño del sistema debe ser lo más sencillo y óptimo posible para agilizar el tiempo de respuesta del mismo.

Rendimiento

- La clasificación de los trabajos periodísticos debe ser lo más eficiente y precisa posible.
- El tiempo de respuesta en la clasificación de trabajos periodísticos debe ser corto, por lo que el procesamiento de los datos se debe efectuar de forma rápida.

Soporte

- El sistema será fácil de probar para facilitar las tareas en la etapa de Prueba.
- El sistema podrá ser extendido, a partir de las características de orientación a objetos de Java y modularidad de NetBeans.

Capítulo 2: Descripción y construcción de la solución propuesta

- Se permitirá la configuración de las tareas por parte del usuario.
- Se ejecutarán tareas en segundo plano en los horarios especificados por el usuario.
- El sistema no se instalará para su uso, solo se copiará donde el usuario estime conveniente.
- El sistema debe dar facilidad de mantenimiento una vez implantada para posibilitar un perfeccionamiento continuo.

Portabilidad

- Las herramientas utilizadas para el desarrollo del sistema son tecnología de software libre y a su vez multiplataforma, lo cual le confiere al sistema esta última característica.

Seguridad

- Sólo los usuarios autorizados podrán acceder al sistema.
- El sistema debe tener protección contra acciones no autorizadas para evitar afectar la integridad de la información almacenada.

Confiabilidad

- Ante cualquier fallo el sistema debe dar una respuesta inmediata.
- El sistema debe posibilitar la recuperación de la información en caso de fallos y/o errores.

Ayuda

- Debe contar con un Manual de Usuario y un sistema de ayuda de forma tal que le brinde orientación al usuario respecto a las opciones con que cuenta el sistema, utilizando textos explicativos que indiquen la acción de estas.

Software

- La máquina computadora debe tener instalado la Máquina Virtual de Java (JVMTM, por sus siglas en inglés, *JavaTM Virtual Machine*) versión 1.6 o superior, que se encuentra en la herramienta de entorno de ejecución de Java (JRETM, por sus siglas en inglés, *JavaTM Runtime Environment*).

Hardware

- Para ejecutar el software los requerimientos mínimos de hardware son: microprocesador Intel Pentium III a 1 GHz de velocidad de procesamiento u otro similar, con 512 MB de memoria RAM y una tarjeta de red.
- El espacio en disco duro dependerá del tamaño del conjunto de documentos a clasificar.
- La máquina computadora debe estar conectada a la red y tener acceso a Internet.

2.3 Análisis, Diseño Conceptual y Arquitectura Técnica

2.3.1 Análisis de Robustez

El análisis de robustez se desarrolla a partir de las descripciones de los casos de uso. Muestra gráficamente la secuencia de acciones de cada caso de uso a través de un diagrama de robustez.

Esta fase de la metodología tiene dos objetivos principales: la desambiguación de las descripciones de los casos de uso, por lo que estas pueden variar; y el descubrimiento de nuevos objetos en el Modelo del Dominio. Esto permite el refinamiento de dicho modelo mediante la identificación de nuevas clases. [89]

A cada caso de uso le corresponde un diagrama de robustez. Los estereotipos del diagrama de robustez son: objeto interfaz (sustantivo), objeto entidad (verbo) y controlador (sustantivo). Los sustantivos pueden relacionarse con verbos y viceversa; y los verbos pueden relacionarse con verbos. [89]

El diagrama de robustez del Caso de Uso Clasificar Noticias Internas, del paquete Clasificación (ver Figura 2.6), es uno de los más importantes y explicativos del sistema; los demás se pueden encontrar en los Anexos [6](#), [7](#), y [8](#).

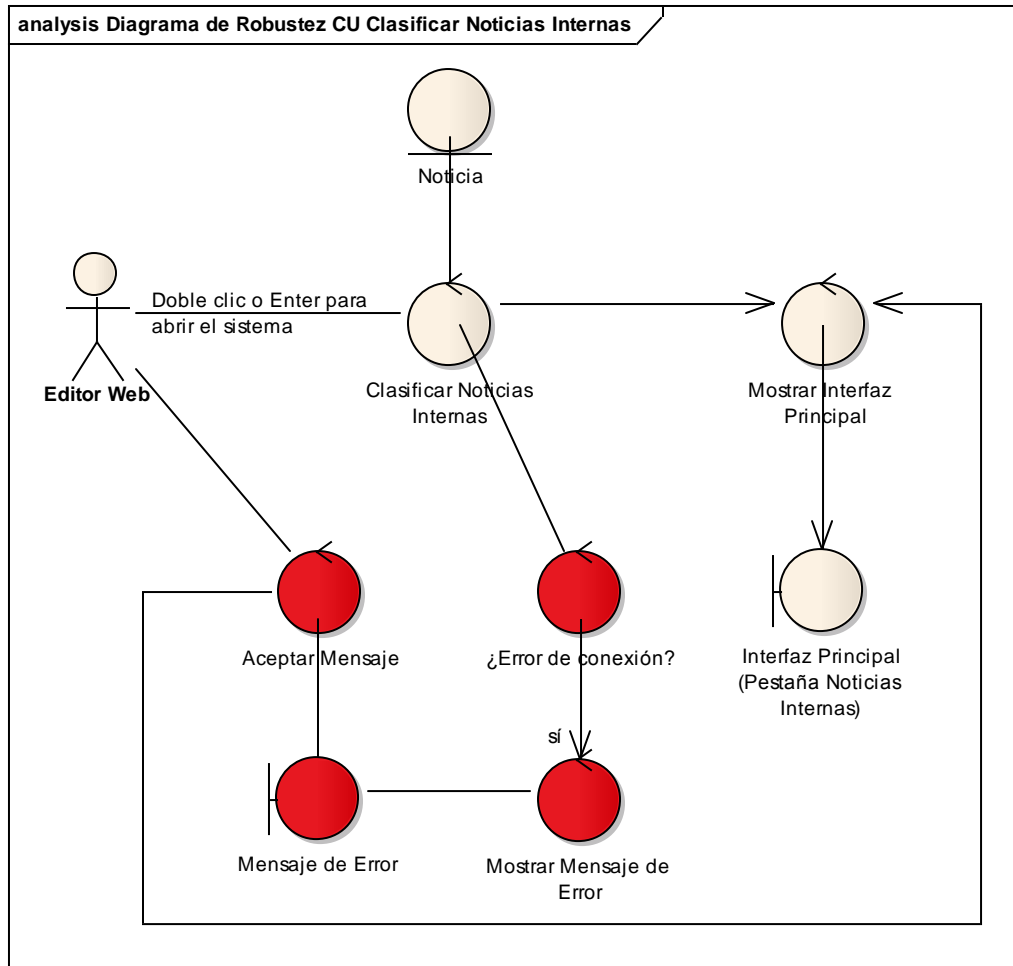


Figura 2.6: Diagrama de Robustez del Caso de Uso Clasificar Noticias Internas del paquete Clasificación.

2.3.2 Arquitectura Técnica

El objetivo de la arquitectura técnica es mostrar una percepción general del producto de software en desarrollo, en cuanto a la arquitectura y las tecnologías utilizadas, así como satisfacer los requerimientos identificados. Se conoce también como arquitectura del sistema y arquitectura del software. [89]

Iconix no propone un método estándar para esta fase de la metodología, sino varios diagramas que se pueden realizar de manera opcional y teniendo en cuenta las características del sistema.

2.3.2.1 Arquitectura de capas

La arquitectura de capas es una metáfora visual por medio de la cual se divide el sistema en varias capas en un diagrama de arquitectura, para mostrar las tecnologías empleadas. La Figura 2.7 muestra un diagrama de arquitectura del sistema mostrando las distintas capas.

Para la presentación del sistema (Vista) se emplea el API Swing de Java, que utiliza los objetos o clases del dominio (POJOs¹⁹) y clases definidas para el acceso a ellos (DAOs²⁰). Se utilizan los APIs Lucene, para el preprocesamiento de las noticias; LingPipe, para la clasificación; y JDBC para el acceso a la BD en MySQL, donde se almacena la información relacionada con el Sitio Web.

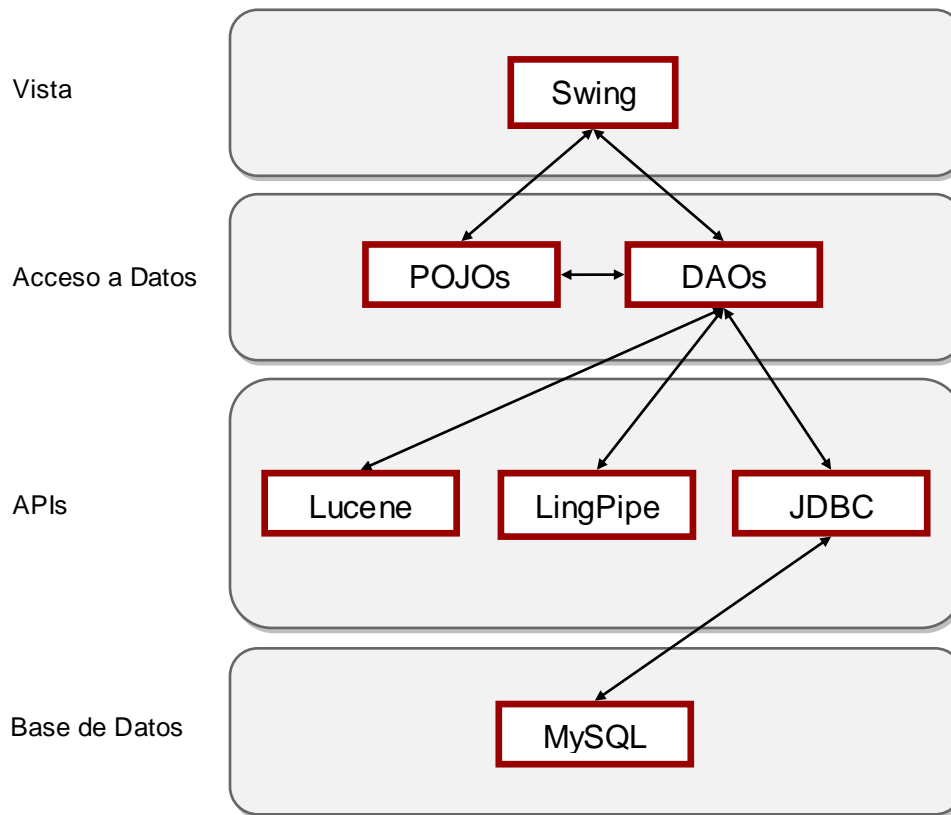


Figura 2.7: Arquitectura del sistema.

¹⁹ Acrónimo de Objetos de Java Viejos y Planos (del inglés *Plain Old Java Objects*).

²⁰ Acrónimo de Objeto de Acceso a Datos (del inglés *Data Access Object*).

2.3.2.2 Modelo de Despliegue

El Modelo de Despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño. Representa una correspondencia entre la arquitectura del software y la arquitectura del hardware.[47] (Ver Figura 2.8)

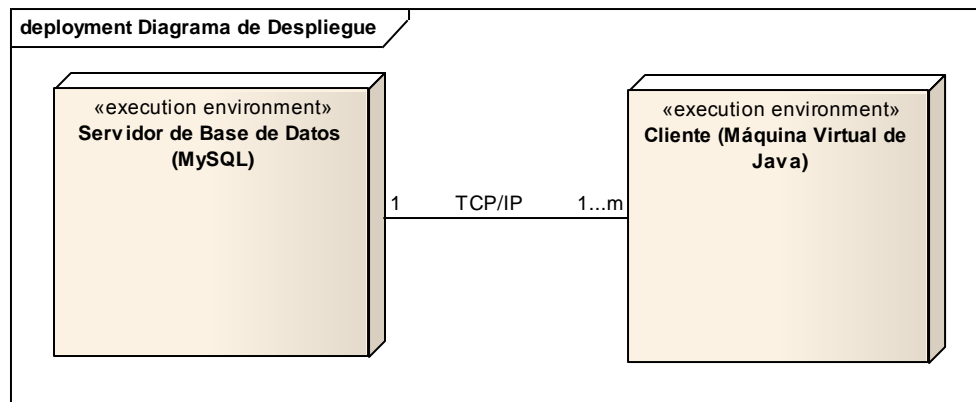


Figura 2.8: Diagrama de despliegue.

La BD “ahora” del Sitio Web se encuentra alojada en Ciudad de La Habana por cuestiones de seguridad. El Sistema de Clasificación Automática de Noticias, por ser una aplicación *desktop*, se encuentra en la computadora cliente para acceder a través de él. La comunicación que se establece entre el servidor de BD y la computadora cliente es a través del protocolo TCP/IP.

2.4 Diseño e Implementación

2.4.1 Diagramas de Secuencia

Los diagramas de secuencia muestran el flujo de actividades del sistema a partir de un diseño más detallado de este, y se realizan uno por cada caso de uso. ICONIX asume que en esta etapa la descripción de los casos de uso son correctos, completos, detallados y explícitos, lo que fue posible con la realización del análisis de robustez. Existe una estrecha relación entre cada caso de uso, su diagrama de robustez y su diagrama de secuencia. [89]

La Figura 2.9 muestra el diagrama de secuencia del Caso de Uso Clasificar Noticias Internas, del paquete Clasificación, uno de los más importantes del sistema, dando continuación a su explicación en el subepígrafe 2.3.1; los restantes se encuentran en el Anexo [9](#), [10](#), [11](#).

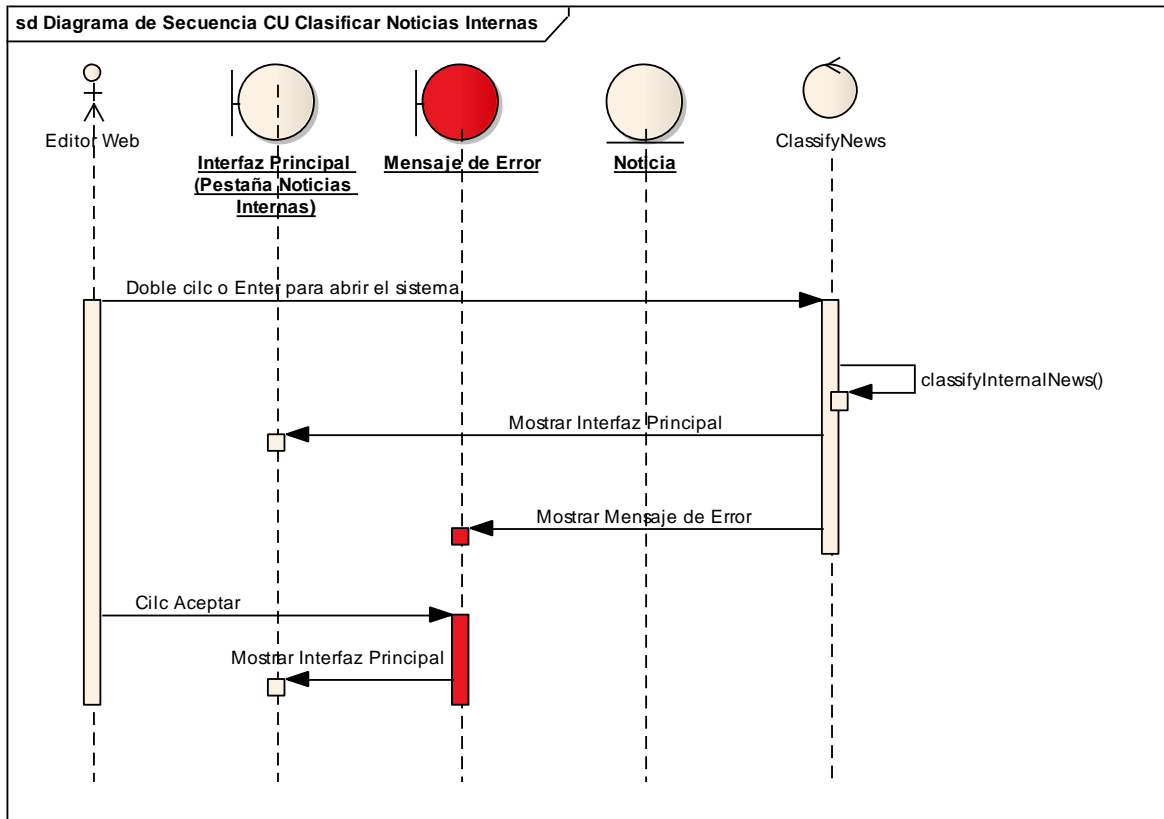


Figura 2.9: Diagrama de Secuencia del Caso de Uso Clasificar Noticias Internas del paquete Clasificación.

2.4.2 Modelo de Clases

El Modelo de Clases define un diagrama que muestra todas las clases del sistema. Iconix propone realizarlo después de los diagramas de secuencia, como una última actualización del Modelo del Dominio con los nuevos objetos o clases identificadas. El diagrama se muestra en el [Anexo 12](#), el cual se dividió en tres partes para su mejor visualización y entendimiento.

2.4.3 Estándar de Código

Para garantizar una implementación uniforme del software, así como facilitar el mantenimiento y la reutilización, se definió un estándar de código, como se muestra en el [Anexo 13](#). Éste se generó mediante el Generador de Estándares StdGenerator versión 1.2, aplicación para la creación y generación de estándares de estilo de códigos a ser utilizados en proyectos de desarrollo de software, creado por el Centro de Referencia de Ingeniería de Software (CRIS), de la Ciudad Universitaria José Antonio Echevarría (CUJAE).

2.5 Prueba

Iconix plantea que la fase de prueba debería comenzarse antes de la implementación. La preparación para la prueba comienza desde la etapa de análisis, identificando los casos de prueba mediante el uso de los diagramas de robustez, los cuales son codificados durante la implementación. La realización de la prueba en etapas tempranas hace posible eliminar gran cantidad de errores, incluso antes de que existan. [89]

Las pruebas están estrechamente relacionadas con los requerimientos, ya que se “prueba” que los mismos se satisfagan. Debe haber al menos una prueba que asegure que cada requerimiento ha sido implementado correctamente. Para ello Iconix propone algunos tipos de prueba y cuándo usarlos. [89]

Se probaron las tecnologías utilizadas antes y luego de la integración. Se realizaron varias pruebas para cada caso de uso, verificando que se diera cumplimiento a cada requerimiento del sistema, para lo que se utilizaron datos reales, es decir, noticias extraídas de la BD del periódico digital.

2.6 Valoración de sostenibilidad

La valoración de sostenibilidad de un producto informático no es más que el proceso de evaluación de impactos ambientales, socio humanistas, administrativos y tecnológicos del mismo, previsibles desde el diseño del proyecto, que favorece su autorregulación, para la satisfacción de la necesidad que

resuelve, con un uso racional de recursos y la toma de decisiones adecuadas a las condiciones del contexto y el cliente.[22]

Con el propósito de favorecer la clasificación de noticias en el proceso editorial del periódico *jahora!* digital en la Casa Editora, se tuvo en cuenta que el sistema informático propuesto como solución fuera sostenible desde las dimensiones administrativa, socio-humanista, ambiental y tecnológico.

Este procedimiento se realiza antes de la etapa de análisis, para conocer tempranamente si es sostenible y factible la solución que se propone antes desarrollarla. Se explica en este epígrafe para no entorpecer el flujo de la metodología de Ingeniería de Software empleada.

2.6.1 Dimensión Administrativa

En la dimensión administrativa se valora si la solución planteada ahorra recursos. Se tienen presente los gastos implicados para desarrollarla e implantarla, la calidad de la producción y los servicios, la administración de recursos y la toma de decisiones administrativas, de manera que se garantice la sostenibilidad administrativa del sistema.

Para el desarrollo del sistema propuesto se estimaron los valores de costo, tiempo y recursos requeridos, para lo que se recurrió al Modelo Constructivo de Costos (COCOMO II, por sus siglas en inglés, *Constructive Cost Model*), que permitió realizar el análisis de factibilidad. La factibilidad de un proyecto esta dada por la determinación de la posibilidad de hacer según restricciones (tiempo, presupuesto, etc.) identificadas y aprobadas teniendo en cuenta los criterios organizativos, económicos, técnicos y de tiempo.

COCOMO II es un modelo que permite estimar el coste, esfuerzo y tiempo asociados a la construcción de un software. Este modelo incluye los siguientes pasos: obtener los puntos de función (UFP, por sus siglas en inglés), estimar la cantidad de instrucciones fuente (SLOC, por sus siglas en inglés) y aplicar las fórmulas de Boehm.[16, 17]

Para obtener los puntos de función es necesario identificar las siguientes características del sistema (ver [Anexo 14](#)): Entradas Externas (EI, por sus siglas

Capítulo 2: Descripción y construcción de la solución propuesta

en inglés), Salidas Externas (EO, por sus siglas en inglés), Consultas Externas (EQ, por sus siglas en inglés), Ficheros Lógicos Internos (ILF, por sus siglas en inglés) y Ficheros de Interfaz Externa (EIF, por sus siglas en inglés).[82] Luego se cuenta la cantidad de funciones de característica por cada nivel de complejidad y se multiplica por el peso asociado en la Tabla 2.1. Todos estos productos se suman y se obtiene la cantidad de puntos de función desajustados. El sistema propuesto no tiene EQ (salidas asociadas al sistema que no tienen elementos de filtraje de información) ni ILF.

Elementos	Simples	Peso	Medios	Peso	Complejos	Peso	Subtotal
EI	1	3	0	4	1	6	9
EO	1	4	1	5	0	7	9
EQ	0	3	0	4	0	6	0
ILF	0	7	0	10	0	15	0
EIF	6	5	0	7	0	10	30
Total	8		1		1		48

Tabla 2.1: Puntos de función desajustados (UFP).

La estimación de la cantidad de líneas de código fuente (SLOC, por sus siglas en inglés) del proyecto se basa en la cantidad de líneas de código por punto de función del lenguaje a usar en la implementación del sistema.[82] Los lenguajes de programación a utilizar son: Java y SQL. Se estima que del empleo total de código, el 95 % es de Java y el 5% de SQL. (Ver Tabla 2.2)

Características	Valor
Lenguaje Java (95%)	$SOLC = UFP * Ratio * Porciento$ $SOLC = 48 * 63 * 95\% = 2872,80$ $KSOLC = 2,8728$
Lenguaje SQL (5%)	$SOLC = UFP * Ratio * Porciento$ $SOLC = 48 * 37 * 5\% = 88,80$ $KSOLC = 0,0888$
KSLOC Total (Líneas de Código Fuente en miles)	$KSOLC(Java) + KSOLC(SQL) = 2,9616$

Tabla 2.2: Cantidad de Líneas de Código Fuente.

Para determinar el esfuerzo asociado al desarrollo del sistema (PM), el tiempo de desarrollo (TDEV) y el costo (CHM), se utilizan los multiplicadores de esfuerzo (EM, por sus siglas en inglés), los factores de escala (SF, por sus siglas en inglés)

Capítulo 2: Descripción y construcción de la solución propuesta

y los valores constantes A, B, C y D (ver Tabla 2.5). Para obtener el valor correspondiente a los multiplicadores de esfuerzo y los factores de escala se le asigna una escala de muy bajo (mayor valor), bajo, nominal, alto, muy alto y extra alto (menor valor), la cual está asociada a valores de acuerdo con las características que se ajustan al producto a desarrollar y al equipo de trabajo (Ver Tabla 2.3 y 2.4).[20, 40]

Factor	Descripción	Escala	Valor
PREC	Precedencia	Bajo	4.96
FLEX	Flexibilidad	Muy Alto	1.01
RESL	Riesgos	Extra Alto	0
TEAM	Cohesión del Equipo	-	-
PMAT	Madurez de las Capacidades	Extra Alto	0
Suma			5.97

Tabla 2.3: Factores de Escala.

Multiplicador	Descripción	Escala	Valor
Del Producto			
RCPX	Confiabilidad y complejidad del producto	Muy Alto	1.91
RUSE	Nivel de reutilizabilidad del desarrollo	Extra Alto	1.24
De la Plataforma			
PDIF	Dificultad de uso de la plataforma	Bajo	0.87
Del Personal			
PERS	Capacidad del personal de desarrollo	Alto	0.83
PREX	Experiencia del personal de desarrollo	Muy Alto	0.74
Del Proyecto			
FCIL	Facilidades de desarrollo	Alto	0.87
SCED	Exigencias sobre el calendario	Bajo	1.14
Producto			1.255186389

Tabla 2.4: Multiplicadores de esfuerzo.

Constante	Valor
A	2.94
B	0.91
C	3.67
D	0.28

Tabla 2.5: Constantes.

El esfuerzo, el tiempo de desarrollo y el costo se calculan a partir de las fórmulas de Boehm, como se muestra en la Tabla 2.6.

Cálculo de	Justificación	Valor
Esfuerzo (hombres/mes)	$E = B + 0.01 * \sum_{j=1}^5 SF_j = 0.9697$ $PM = A * KSLOC^E * \prod_{i=1}^7 EM_i$	10,57534886 \approx 11
Tiempo de Desarrollo (meses)	$F = D + 0.2 * \sqrt{E - B} = 0.29194$ $TDEV = C * PM^F$ <p><i>CH Estim. (Cantidad de Hombres Estimado)</i></p> $CH Estim. = PM / TDEV = 1,447424912$ <p>\approx 2 hombres</p> <p><i>CH Real (Cantidad de Hombres Real) = 1</i></p> $TDEV Real = PM / CH Real = 10,57534886$ <p>\approx 11 meses</p>	7,306319503 \approx 8
Costo (pesos)	<p><i>CHM (Costo por Hombres/Mes)</i></p> $CHM = CH Real * Salario Promedio (300)$ $CHM = \$300$ $COSTO = CHM * PM$	\$ 3.172,60

Tabla 2.6: Esfuerzo, Tiempo de desarrollo y Costo.

El Tiempo de Desarrollo se calcula considerando que un trabajador trabaja al mes 152 horas. El desarrollo del sistema informático se estimó que duraría

Capítulo 2: Descripción y construcción de la solución propuesta

aproximadamente 8 meses, realizándolo con 2 personas. Pero como solo se cuenta con una, se determinó que el tiempo aproximado de terminación del sistema sería 11 meses, con un costo de \$ 3.172,60.

A pesar del costo estimado para desarrollar el sistema informático propuesto, no se incurre en gastos debido a que la desarrolladora es una estudiante. Además, los beneficios que éste brindará al proceso de clasificación de noticias en el proceso editorial del periódico *jahora!* digital son considerables.

Con la implantación del sistema informático disminuirá el tiempo de gestión de noticias por la facilidad que proporcionará el mismo al realizar algunas tareas automáticamente. Por otra parte, permitirá realizar una búsqueda de noticias con la calidad requerida por parte del usuario, debido a que el sistema clasificará las mismas sin ambigüedades.

Uno de los resultados que genera el corto tiempo empleado anteriormente aludido, es la disminución del gasto de energía eléctrica, en sintonía con el ahorro de energía que lleva a cabo el país.

Las herramientas utilizadas para el desarrollo del sistema en su totalidad son libres, por lo que no se incurre en gastos para desarrollar y aplicar el proyecto.

A partir de lo antes analizado y los beneficios que proporciona el sistema, se arribó a la conclusión que éste es sostenible desde la dimensión administrativa.

2.6.2 Dimensión Socio-Humanista

En la dimensión socio-humanista se evalúa cómo el sistema propuesto contribuye con el desarrollo del modo de vida de un grupo social y con la formación ético-humanista de los gestores del proyecto informático, y si satisface una necesidad social. El fortalecimiento del factor humano es necesario para mejorar el rendimiento en los servicios. La automatización de los procesos proporciona el bienestar de los trabajadores en el desempeño laboral a partir de las comodidades brindadas.

El sistema informático propuesto resolverá la necesidad planteada en la Casa Editora, pues facilitará una búsqueda cómoda y una clasificación precisa de

Capítulo 2: Descripción y construcción de la solución propuesta

noticias y disminuirá el tiempo que se emplea en esta tarea actualmente, con lo que se contribuirá a la calidad de la misma en el proceso de edición.

Con el uso del sistema, la carga de trabajo se reducirá, al realizar algunas de las tareas automáticamente. Estas mejoras en las condiciones de trabajo darán lugar a una gran satisfacción del personal implicado de la Casa Editora, lo que favorecerá una mejor calidad en el proceso productivo. Se garantizará que la entrega de los resultados sea correcta, segura y en un tiempo breve.

El sistema podrá ser generalizado, pues el problema que resuelve no es sólo de la Casa Editora de la provincia de Holguín, por lo que podrá ser adaptado para cualquier institución análoga del país. Por tales razones, dentro de las concepciones del sistema se tuvieron en cuenta su flexibilidad y versatilidad, para capturar las generalidades, pero también las posibles particularidades que pueden tener las organizaciones de este ámbito o dominio. La naturaleza modular y extensible de la tecnología con que se desarrolla el sistema hace que no constituya un problema la extensión u evolución del mismo.

Se tuvo en cuenta el rechazo al cambio que podía surgir una vez que se implantara el sistema, lo cual era normal, debido a la tendencia del ser humano a hacer costumbre de lo cotidiano y rutinario y a la resistencia inconsciente ante los cambios de su entorno. Para favorecer la aceptación del sistema se llevaron a cabo entrevistas con los usuarios finales, para explicarles en profundidad las ventajas que el sistema proporcionaría y cómo el sistema podía serles fácil y cómodo de usar.

El sistema se desarrolló en una interfaz con ambiente *desktop*. El flujo de trabajo inmerso en el sistema será similar a como se lleva a cabo en la Casa Editora; es decir, lo más intuitivo posible y fácil de manipular, de tal forma que los usuarios no se pierdan mientras trabajaban sobre él y no lo rechacen ante el cambio.

El sistema hizo aportes a la ciencia y la tecnología, ya que no existen otras herramientas de clasificación de noticias para los periódicos digitales cubanos, facilitando así las tareas en este sentido.

A partir de lo analizado anteriormente, se arribó a la conclusión de que el sistema es sostenible desde la dimensión socio-humanista.

2.6.3 Dimensión Ambiental

En la dimensión ambiental se valora si el sistema resulta favorable o no para las personas o cosas y cómo el mismo minimiza daños e impactos. Cada vez más, el derroche excesivo se incrementa a partir de soluciones no razonables ambientalmente. Por lo que resulta muy importante que toda solución de un problema no repercuta en daños y alteraciones al medio ambiente.

En la Casa Editora las condiciones de las estaciones de trabajo no son las mejores, debido a la ausencia de protectores de pantallas, asientos cómodos y regulables respecto a la posición que se encuentra el monitor y la ubicación correcta de éste, lo cual puede repercutir en daños en la cervical y la columna, estrés y cansancio en la vista de los usuarios.

Teniendo en cuenta lo antes expuesto y las largas horas que los usuarios se hallan frente a las computadoras durante el flujo de trabajo, no se hará uso de colores agresivos a la vista en la interfaz del sistema, sino de los de tonalidades claras que se encuentran en la gama del negro, gris, rojo y blanco, en concordancia con los colores del periódico digital y en busca de un efecto atractivo y agradable en la comunicación entre el sistema y los usuarios.

Por otra parte, se tuvieron en cuenta las exigencias fisiológicas del ser humano al seleccionar la tipografía, el tamaño de letra y el espaciamiento entre caracteres, para la cómoda visualización de los contenidos, alineación y tamaño de las imágenes. Todo esto ayuda a evitar el cansancio visual, los daños en la columna y el estrés de los usuarios al minimizar el tiempo frente al monitor, por lo que se favorece la calidad en las labores productivas.

Se podrán reutilizar los componentes y recursos del sistema, debido a las características de modularidad de las tecnologías empleadas y a que se tuvo en cuenta la generalidad en el diseño del mismo, aunque respetando las particularidades específicas de la Casa Editora *jahora!*

A partir de lo analizado anteriormente se arribó a la conclusión de que el sistema es sostenible desde la dimensión ambiental.

2.6.4 Dimensión Tecnológica

En la dimensión tecnológica se evalúa si la tecnología usada es adecuada y asimilable con el usuario.

Para el empleo del sistema los usuarios se encuentran capacitados y no necesitan de preparación informática, debido a que hacen uso de computadoras para su labor diaria; además el sistema será intuitivo y fácil de usar y poseerá una Ayuda que facilitará su manipulación.

En cuanto a la infraestructura electrónica, la Casa Editora cuenta con los recursos precisos para la implantación y aplicación del sistema. Las características que poseen las computadoras utilizadas por los trabajadores cumplen con los requerimientos necesarios para hacer uso del sistema y además se encuentran conectadas a la red.

Con el fin de facilitar el mantenimiento del sistema, se definió un estándar de código, se describieron con comentarios las funciones fundamentales y de forma general lo que hacen las clases, además, se le pusieron nombres intuitivos para proporcionar una mejor comprensión y entendimiento.

Por otra parte, el sistema permitirá su evolución en el tiempo, debido a la flexibilidad que proporcionará (explicado anteriormente en la dimensión socio-humanista). Además, permitirá cambios, ya sea de mejoras de hardware, red e incluso de plataforma.

A partir de lo analizado anteriormente se arribó a la conclusión de que el sistema es sostenible desde la dimensión tecnológica.

Conclusiones del capítulo

En este capítulo se determinaron, a través de la metodología ICONIX, los objetos o entidades fundamentales implicadas en el dominio del problema y la relación existente entre ellas. Se identificaron las exigencias que debe cumplir la solución propuesta, así como los usuarios finales y las acciones que estos deben ejecutar. Además, se especificaron los cursos básicos y alternos de las acciones del sistema, así como los procesos de diseño, implementación y prueba de la

Capítulo 2: Descripción y construcción de la solución propuesta

metodología utilizada. Se definió, además, un estándar de código para la implementación.

Se realizó un estudio de la sostenibilidad administrativa, socio-humanista, ambiental y tecnológica del sistema, por lo que se arriba a la conclusión de que su desarrollo es factible, sostenible y perdurable en el tiempo.

Conclusiones

El diagnóstico profundo llevado a cabo en la Casa Editora *¡ahora!* permitió detectar las deficiencias en la clasificación de noticias en el proceso editorial del periódico *¡ahora!* digital, lo que constituyó el punto de partida de la investigación.

La metodología de desarrollo de software empleada para realizar el diagnóstico del problema y modelar la solución, resultó fundamental para maximizar los índices de calidad en los procesos de ingeniería de software implicados.

Valoradas las dimensiones administrativa, socio-humanista, ambiental y tecnológica del sistema, se puede afirmar que el producto final es sostenible y perdurable en el tiempo.

La combinación adecuada de las tecnologías utilizadas facilitó el desarrollo del Sistema de Clasificación Automática de Noticias potenciando la funcionalidad del mismo.

A partir de la evaluación de los algoritmos de clasificación, basada en las pruebas estadísticas de Friedman y Wilcoxon, así como las medidas de evaluación empleadas, se decidió utilizar en el Sistema de Clasificación Automática de Noticias el algoritmo *Dynamic Language Model*, de acuerdo al buen desempeño mostrado en todo momento.

Los procesos relacionados con las noticias externas (descarga y clasificación) no fueron automatizados debido a que resultaron complejos para el tiempo disponible.

El Sistema de Clasificación Automática de Noticias favorece la clasificación de noticias en el proceso editorial del periódico *¡ahora!* digital, dando cumplimiento al objetivo trazado en la investigación.

Recomendaciones

- Incorporar al Sistema de Clasificación Automática de Noticias las noticias extraídas de fuentes periodísticas externas cubanas.
- Aplicar otras técnicas de reducción de la dimensionalidad, con el fin de favorecer la efectividad de la clasificación, así como disminuir el costo en tiempo.
- Incorporar a las pruebas estadísticas algoritmos de clasificación que no se encuentran implementados en el API Lingpipe.
- Incorporar al periódico *jahora!* digital funcionalidades que permitan la clasificación múltiple de noticias.
- Incorporar categorías relacionadas con los géneros periodísticos para favorecer la discriminación entre documentos por categorías a clasificar.
- Perfeccionar la interfaz basada en opiniones de los usuarios finales.

Glosario de Términos

Base de datos: Una base de datos consta de una colección de tablas que contienen datos y otros objetos como vistas, índices, procedimientos almacenados y desencadenadores, que se definen para poder llevar a cabo distintas operaciones con datos. Los datos almacenados en una base de datos suelen estar relacionados con un tema o un proceso determinados como, por ejemplo, la información de inventario para el almacén de una fábrica.

CMS: Acrónimo de *Content Management System* (Sistema de Gestión de Contenido). Sistema que facilita la gestión de contenidos Web en todos sus aspectos: creación, mantenimiento, publicación y presentación.

Corpus: Una colección de lenguaje humano, mayormente provista con características lingüísticas y/o de contenido.

GUI: Acrónimo de *Graphics User Interface* (Interfaz Gráfica de Usuario). Parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

Hardware: Conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

HTML: Acrónimo de *Hyper Text Markup Language* (Lenguaje de Marcado de HiperTexto). Lenguaje de programación para diseñar páginas Web. Básicamente HTML consta de texto plano (ASCII) y una serie de etiquetas (Tags).

IDE: Acrónimo de *Integrated Development Environment* (Entorno de Desarrollo Integrado). Entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

PDF: Acrónimo de *Portable Document Format* (Formato de Documento Portable). Tipo de fichero de texto.

RSS: Canal de noticias en formato XML que permite publicar artículos simultáneamente en diferentes medios a través de una fuente a la que pertenece.

Software: Conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

UML: Acrónimo de *Unified Modeling Language* (Lenguaje Unificado de Modelado). Notación estándar o lenguaje de propósito general para modelar objetos del mundo real, como primer paso en el desarrollo de programas orientados a objetos.

XML: Acrónimo de *eXtensible Markup Language* (Lenguaje de Mercado Extensible). Lenguaje utilizado en la actualidad para intercambio de información en un formato entendible para distintas aplicaciones.

Bibliografía

- [1] *Aprendizaje bayesiano*. Departamento de Sistemas Informáticos y Programación. Universidad Complutense de Madrid.
- [2] *Fundamentos e Historia del Periodismo*. Consultado: 4 de junio del 2009. Disponible en: <http://html.rincondelvago.com/fundamentos-e-historia-del-periodismo.html>.
- [3] *ICONIX*. Consultado: 12 de mayo del 2010. Disponible en: http://apuntes.rincondelvago.com/modelamiento-de-datos_iconix.html.
- [4] *Weka 3: Data Mining Software in Java*. Consultado: 25 de enero del 2010. Disponible en: <http://www.cs.waikato.ac.nz/ml/weka/>.
- [5] (2008, 10 de marzo). *Internet: Surgimiento y evolución*. *Periodismo digital 2008*. Consultado: 11 de enero del 2010. Disponible en: <http://pdigital2008.blogspot.com/2008/03/prueba-1.html>.
- [6] (2008). *Introducción a Apache Lucene (Java)*. Consultado: 17 de febrero del 2010. Disponible en: <http://es.debugmodeon.com/articulos/introduccion-a-apache-lucene-java.htm>.
- [7] (2009). *Canales RSS*. Consultado: 6 de julio del 2009. Disponible en: <http://www.oas.org/documents/spa/rss.asp>.
- [8] (2009). *Sistemas de Información. Tema 6: Métodos, lenguajes y paradigmas*. Consultado: 10 de septiembre de 2009. Disponible en: <http://kybele.escet.urjc.es/documentos/SI/%5BSI-2006-07%5DT6 Metodos Lenguajes Paradigmas.pdf>.
- [9] (2009). *Welcome to Lucene!* The Apache Software Foundation.
- [10] (2010). *Manifiesto Ágil*. Consultado: 9 de septiembre del 2010. Disponible en: <http://agilemanifesto.org/>.
- [11] (2010). *Modelo del dominio*. Consultado: 12 de mayo del 2009. Disponible en: http://www.worldlingo.com/ma/enwiki/es/Domain_model.htm.
- [12] AHONEN MYKA, H. (2002). *Discovery of Frequent Word Sequences in Text Source*. Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery, London, UK.
- [13] ALIAS-I, INCORPORATED. (2009). *LingPipe API*. Consultado: 20 de noviembre del 2009. Disponible en: <http://alias-i.com/lingpipe/docs/api/index.html>.
- [14] ARCO GARCÍA, L. (2008). *Agrupamiento basado en el concepto de intermediación diferencial y la aplicación de la teoría de los conjuntos aproximados para valorar resultados de agrupamientos*. Tesis en opción del grado científico de Doctor en Ciencias Técnicas. Departamento de Ciencia de la Computación. Universidad Central "Marta Abreu" de Las Villas. Santa Clara.
- [15] BLANCO, C.D.P. (2010). *Panorama Mundial en el uso del Software de Código Abierto. Perspectivas en Cuba. III Taller de Informatización de la Sociedad*. Holguín, Oficina para la Informatización.
- [16] BOEHM, B.W., B. CLARK, E. HOROWITZ, et al. (1995). *The COCOMO 2.0 Software Cost Estimation Model*. Disponible en: <http://sunset.usc.edu/COCOMOII/cocomo.html>.

- [17] BOEHM, B.W., B. CLARK, E. HOROWITZ, et al. (1995) *Cost Models for Future Software Life Cycle Processes: COCOMO II*. Annals of Software Engineering Special Volume on Software Process and Product Measurement 1: 45-60
- [18] BORDIGNON, F. (2007) *Clasificación de textos por el método Naive Bayes*. Blog: Apuntes, son solo apuntes
- [19] CASA EDITORA AHORA (2008). *Perfil Editorial de Ahora*.
- [20] CENTER FOR SOFTWARE ENGINEERING, USC (2000). *COCOMO II. Model Definition Manual Version 2.1*.
- [21] COHEN, W. & Y. SINGER (1999). *Context-sensitive learning methods for text categorization*. ACM Transactions on Information Systems 17(2): ACM Transactions on Information Systems.
- [22] CONCEPCIÓN GARCÍA, M.R. (2006). *Estrategia para desarrollar la gestión ambiental de proyectos informáticos sostenibles en la formación del Ingeniero Informático*. Tesis presentada en opción al Título Académico de Master en Gestión Ambiental. Instituto Superior de Tecnología y Ciencias Aplicadas.
- [23] CHAKRABARTI, S. (2003). *Mining the Web: discovering knowledge from hypertext data*. Morgan Kaufmann, San Francisco, CA, cop.
- [24] CHASKI, C. (2005). *Who's at the Keyword? Authorship Attribution in Digital Evidence Investigations*. International Journal of Digital Evidence 4(1).
- [25] CHRISTOPHER D., M., P. RAGHAVAN & H. SCHÜTZE *An Introduction to Information Retrieval*.
- [26] DAVIS D., L. & M. RINGUETTE (1994). *A comparison of two learning algorithms for text categorization*. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US.
- [27] DEL VALLE GASTAMINZA, F. (2004). *Lenguajes documentales: los tesauros*. Consultado: 12 de junio del 2009. Disponible en: <http://www.ucm.es/info/multidoc/prof/fvalle/tesauro.htm>.
- [28] DÍAZ (2009). *Algoritmo de votación incremental INC-ALVOT para clasificación supervisada*. Facultad de Ingeniería. Universidad Antioquia 50: 195-204.
- [29] DOAN, S. (2005). *A Fuzzy-Based Approach for Text Representation in Text Categorization*. IEEE International Conference on Fuzzy Systems.
- [30] ECHEVERRÍA, J. (1998). *Internet y el periodismo electrónico*. Consultado: 4 de junio del 2009. Disponible en: <http://saladeprensa.org/art08.htm>
- [31] FERNÁNDEZ, A.Q. (2007). *Clasificación Automática de Textos mediante Reglas de Asociación*. Facultad de Matemática y Computación. Universidad de La Habana.
- [32] FERRI, C. *Mi página de Weka*. Consultado: 25 de enero del 2010. Disponible en: <http://users.dsic.upv.es/~cferri/weka/>.
- [33] FIGUEROLA, C.G., J.L. ALONSO BERROCAL, A.F. ZAZO RODRÍGUEZ, et al. (2002). *Algunas Técnicas de Clasificación Automática de Documentos*.

- [34] FIGUEROLA, C.G., Á.F. ZAZO RODRÍGUEZ & J.L. ALONSO BERROCAL (2001). *Automatic vs. manual categorization of documents in Spanish*. Journal of Documentation 57(6): 763–773.
- [35] FOWLER, M. (2003, abril de 2003). *La Nueva Metodología*. Consultado: 3 de diciembre del 2008. Disponible en: <http://martinfowler.com/articles/newMethodology.html>.
- [36] FRANCO ARCEGA, A., J.A. C.O., G. SÁNCHEZ DÍAZ, et al. (2008). *Árboles de Decisión para Grandes Conjuntos de Datos*. Coordinación de Ciencias Computacionales INAOE.
- [37] FRIEDMAN, M. (1937). *The use of ranks to avoid the assumption of normality implicit in the analysis of variance*. Journal of the American Statistical Association 32: 675–701.
- [38] FRIEDMAN, M. (1940). *A comparison of alternative tests of significance for the problem of m rankings*. Annals of Mathematical Statistics: 86–92.
- [39] GENKIN, A., D.D. LEWIS & D. MADIGAN *Large-Scale Bayesian Logistic Regression for Text Categorization*.
- [40] GÓMEZ, A., M.D.C. LÓPEZ, S. MIGANI, et al. (1997). *COCOMO - Un modelo de estimación de proyectos de software*.
- [41] GONZÁLEZ CID, J.J., J. R.M. & R.F. F.R. (2005). *Modelos Anti-Spam de Inteligencia Artificial*. Conferencia IADIS Ibero-Americana WWW/Internet.
- [42] GÖVERT, N., M. LALMAS & N. FUHR (1999). *A probabilistic description-oriented approach for categorising Web documents*. Proceedings of the Eighth International Conference on Information and Knowledge Management, New York, ACM.
- [43] HORSTMANN, C.S. & G. CORNELL (2004). *Core Java™ 2*. Prentice Hall PTR.
- [44] HUETE GUADIX, J.F. & J.M. FERNÁNDEZ LUNA (2008). *Agrupamiento y clasificación documental*. Recuperación de Información. Universidad de Granada.
- [45] HUETE GUADIX, J.F. & J.M. FERNÁNDEZ LUNA (2008). *Indexación de documentos*. Recuperación de Información. Universidad de Granada.
- [46] HUMPHREY, W.S. (2001). *Introducción al Proceso Software Personal*. Madrid, Addison Wesley.
- [47] JACOBSON, I. & G. BOOCH (2000). *El Proceso Unificado de Desarrollo del Software*. Addison-Wesley.
- [48] JARDINES MÉNDEZ, J.B. (2007, 18 de mayo). *Canales RSS*. Consultado: 6 de julio del 2009. Disponible en: <http://www.uvirtual.sld.cu/canales-rss.htm>.
- [49] JOACHIMS, T. (1997). *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization*. Proceedings of the 14th International Conference on Machine Learning, Nashville, U.S., Morgan Kaufmann Publishers.
- [50] JOACHIMS, T. (1998). *Text Categorization with Support Machines: Learning with Many Relevant Features*. Proceedings of the 10th European Conference on Machine Learning, Lecture Notes in Computer Science.

- [51] JOACHIMS, T. (2001). *A statistical learning model of text classification with support vector machines*. Proceedings of the 24th ACM International Conference on Research and Development in Information Retrieval, ACM Press.
- [52] KASTER, A., S. SIERSDORFER & G. WEIKUM (2005). *Combining Text and Linguistic Document Representations for Authorship Attribution*. Workshop Stylistic Analysis of Text for Information Access 28: 27-35.
- [53] KHARE, R., D. CUTTING, K. SITAKER, et al. (2005). *Nutch: A Flexible and Scalable Open-Source Web Search Engine*. Chiba, Japan.
- [54] KIVINEN, J., M. WARMUTH & P. AUER (1995). *The perceptron algorithm vs. winnow: Linear vs. logarithmic mistake bounds when few input variables are relevant*. Conference on Computational Learning Theory.
- [55] KOHAVI, R. & J.R. QUINLAN (2002). *Decision-tree discovery*. Oxford University Press.
- [56] KOHONEN, T., S. KASKI, K. LAGUS, et al. (2000). *Self organization of a massive document collection*. *Neural Networks* 11(3): 574–585.
- [57] KONCHADY, M. (2008). *Building Search Applications*. Lucene, LingPipe and Gate. Oakton, Virginia, Mustru Publishing.
- [58] LANDAUER, T.K. & S.T. DUMAIS (1997). *A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge*. *Psychological Review* 104(2): 211-240.
- [59] LEMUR PROJECT. (2007, 21 de junio). *Lemur Project Tutorials: Starting Out. Overview. Overview of the Lemur Toolkit*. Consultado: 20 de noviembre del 2009. Disponible en: http://www.lemurproject.org/tutorials/begin_overview-1.php.
- [60] LEMUR PROJECT. (2009). *Lemur Project Documentation*. Consultado: 20 de noviembre del 2009. Disponible en: <http://www.lemurproject.org>.
- [61] LEMUR PROJECT. (2009, 21 de diciembre). *The Lemur Toolkit for Language Modeling and Information Retrieval*. Consultado: 20 de noviembre del 2009. Disponible en: <http://www.lemurproject.org>.
- [62] LEMUR PROJECT. (2009). *README. Indri 2.6*. Consultado: 20 de noviembre del 2009. Disponible en: <http://www.lemurproject.org>.
- [63] LEWIS, D. (1991). *Evaluating text categorization*. Proceedings of the Speech and Natural Language Workshop, Asilomar, CA.
- [64] LEWIS, D.D. & M. RINGUETTE (1994). *A comparison of two learning algorithms for text categorization*. Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, US.
- [65] LEWIS, D.D., R.E. SCHAPIRE, J.P. CALLAN, et al. (1996). *Training algorithms for linear text classifiers*. Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland (Special Issue of the SIGIR Forum), ACM.
- [66] LIN, X., D. SOERGEL & G. MARCHIONINI (1991). *Self-organizing semantic map for information retrieval*. ACM.

- [67] MARON, M. (1961). *Automatic indexing: an experimental inquiry*. Journal of the ACM 8: 404–417.
- [68] MARTÍN GARCÍA, M. (2009). *Sistema de clasificación automática de críticas de cine*. Proyecto Fin de Carrera Ingeniería Superior de Telecomunicación. Escuela Politécnica Superior. Universidad Carlos III de Madrid. Madrid.
- [69] MCCALLUM, A. & K. NIGAM (1998). *A Comparison of Event Models for Naive Bayes Text Classification*. AAAI Workshop on Learning for Text Categorization: 41–48.
- [70] METZLER, D., T. STROHMAN, H. TURTLE, et al. (2004) *Indri at TREC 2004: Terabyte Track*.
- [71] MITCHELL, T.M. (1996). *Machine Learning*. New York, McGraw Hill.
- [72] MOENS, M.-F. & J. DUMORTIER (1999). *Automatic categorization of magazine articles*. Conferentie Informatiewetenschap 1999. P.d. Bra & L. Hardman. Amsterdam.
- [73] NAVARRO ZAMORA, L. (2001, Agosto). *Los periódicos on line: sus características, sus periodistas y sus lectores*. Sala de Prensa Consultado: 4 de junio del 2009. Disponible en: <http://www.saladeprensa.org/art253.htm>.
- [74] NIGAM, K., A. MCCALLUM, S. THRUN, et al. (2000). *Text classification from labelled and unlabeled documents using EM*. Machine Learning 39(2/3): 103-134.
- [75] OAKS, S. & H. WONG (2004). *Java Threads*. O'Reilly.
- [76] ORWIG, C., R. CHEN & H. SCHUFFELS (1996). *Internet categorization and search: a machine learning approach*. Journal of Visual Communications and Image Representation 1(7): 88–102.
- [77] OUNIS, I., G. AMATI, V. PLACHOURAS, et al. (2004). *Terrier: A High Performance and Scalable Information Retrieval Platform*. Department of Computing Science. University of Glasgow Scotland, UK.
- [78] ÖZGÜR, A. (2002). *Supervised and Unsupervised Machine Learning Techniques for Text Document Categorization*. B.S. in Computer Engineering, Advisor: Ethem Alpaydın. Boğaziçi
- [79] PATRICIA DE SAN MARTIN OLIVA, C.R. (2010, 24 de mayo). *Uso de Metodología ICONIX*. Consultado: 5 de junio del 2010. Disponible en: <http://www.buenastareas.com/ensayos/Uso-De-Iconix/344458.html>.
- [80] PAUMIER SAMÓN, R., Y. PÉREZ VILLAZÓN & A. MENESES ABAD (2007). *Guía Cubana para la migración a software libre*.
- [81] PEDREIRA, J. (2004, Febrero). *La verdadera historia del origen de Internet*. Consultado: 11 de enero del 2010. Disponible en: http://www.sitiosargentina.com.ar/notas/Febrero_2004/89.htm.
- [82] PERALTA, M. (2004) *Estimación del esfuerzo basada en casos de uso*.
- [83] PROYECTO ECLIPSE. *Overview*. Consultado: 17 de febrero del 2010. Disponible en: <http://eclipse.org/whitepapers/eclipse-overview.pdf>.
- [84] PROYECTO LUCENE. *Nutch*. Consultado: 23 de marzo de 2010. Disponible en: <http://lucene.apache.org/nutch/>.

- [85] QUANTITATIVE SOFTWARE MANAGEMENT, I. (2002, Julio). *QSM Function Point Programming Languages Table* Consultado: 9 de octubre del 2008. Disponible en: <http://www.qsm.com/FPGearing.html>.
- [86] QUINLAN, J.R. (1993). *Programs for Machine Learning*. Morgan Kaufmann, San Mateo, C.A.
- [87] QUINTANA, D. (2007, 11 de diciembre). *Archive for category Lucene*. Consultado: 17 de febrero del 2010. Disponible en: <http://blog.darioquintana.com.ar/category/lucene/>.
- [88] ROCCHIO, J.J. (1971). *Relevance feedback in information retrieval. The SMART Retrieval System: Experiments in Automatic Document Processing*. Englewoods Cliffs, New York, Prentice-Hall.
- [89] ROSENBERG, D. & M. STEPHENS (2007). *Use Case Driven Object Modeling with UML. Theory and Practice*. USA.
- [90] ROSENBERG, D. & M. STEPHENS (2007). *Use Case Driven Object Modeling with UML: Theory and Practice*. USA.
- [91] RUIZ SHULCLOPER, J. (1990). *Modelos Matemáticos para el Reconocimiento de Patrones*.
- [92] RUIZ SHULCLOPER, J. & M. LAZO CORTÉS (1999). *Mathematical algorithms for the supervised classification based on fuzzy partial precedence*. *Mathematical and Computer Modeling* 29: 111-119.
- [93] SALGADO, B. & V. CAMARGO. (2009, 14 de diciembre). *Sun Microsystems lanza NetBeans IDE 6.8*. Disponible en: <http://es.sun.com/sunnews/press/20091214.jsp.htm>.
- [94] SALTON, G. & C. BUCKLEY (1988). *Term weighting approaches in automatic text retrieval*. *Information Processing and Management* 24: 513–523.
- [95] SCHUTZE, H., D.A. HULL & J.Q. PEDERSEN (1995). *A comparasion of classifers and document representations for the routing problem*.
- [96] SEBASTIANI, F. (2002). *Machine Learning in Automated Text Categorization*. *ACM Computing Surveys* 34(1): 1-47.
- [97] SERRADILLA GARCÍA, F. (2006). *Indexación con Lucene*. *Ingeniería de Sistemas y Automática*, Departamento de Sistemas Inteligentes Aplicados.
- [98] SHESKIN & J. DAVID (2004). *Parametric and Nonparametric Statistical Preocedures*
- [99] STALLMAN, R.M. (2004). *Software libre para una sociedad libre*. Traficantes de Sueños.
- [100] STROHMAN, T. (2005) *Dynamic Collections in Indri*.
- [101] STROHMAN, T., D. METZLER, H. TURTLE, et al. (2004) *Indri: A language-model based search engine for complex queries*
- [102] THE UNIVERSITY OF SHEFFIELD. (2010). *GATE is...* Consultado: 12 de marzo del 2010. Disponible en: <http://gate.ac.uk/index.html>.
- [103] TOPLEY, K. (1999). *Core SWING advanced programming*. Prentice Hall PTR
- [104] VALERO, A.T. (2005). *Extracción de Información con Algoritmos de Clasificación*. Tesis sometida como requisito parcial para obtener el grado

- de Master en Ciencias en la especialidad de Ciencias Computacionales. Instituto Nacional de Astrofísica, Óptica y Electrónica. Tonantzintla, Pue.
- [105] VAN RIJSBERGEN, C.J. (1979). *Information Retrieval*. Butterworths, London.
 - [106] VAPNIK, V. (1999). *The nature of Statical Learning Theory*. Ed. Springer.
 - [107] VÁZQUEZ ACOSTA, M. (2009). *Programa de Informatización de la Prensa*.
 - [108] VILLASEÑOR PINEDA, L., M.M. Y.G., A. LÓPEZ LÓPEZ, et al. (2003). *Recopilación y estructuración automática de contenidos educativos digitales a partir de la Web*.
 - [109] YANG, Y. (1999). *An evaluation of statistical approaches to text categorization*. Information Retrieval 1(1-2): 69-90.
 - [110] YANG, Y. & X. LIU (1999). *A re-examination of text categorization methods*. 22nd Annual International SIGIR. Berkley: 42–49.
 - [111] YANG, Y. & J.O. PEDERSEN (1997). *A comparative study on feature selection in text categorization*. Journal of Artificial Intelligence Research 6: 1-34.
 - [112] ZIPF, G.K. (1949). *Human Behaviour and the Principle of Least Effort*. Addison-Wesley.

Anexos

Anexo 1. Lista de palabras vacías o stop words utilizada

a, ante, antes, bajo, cabe, con, contra, de, desde, en, entre, hacia, hasta, para, por, según, sin, sobre, tras, y, e, ni, o, u, pero, empero, mas, sino, porque, conque, puesto, que, pues, si, aunque, tan, como, conforme, cuando, mientras, luego, así, quien, cual, cuyo, donde, cuanto, aquí, allí, ahí, acá, allá, cerca, lejos, arriba, abajo, encima, debajo, detrás, enfrente, fuera, dentro, ayer, anteayer, entonces, ya, hoy, ahora, después, mañana, aún, todavía, siempre, nunca, jamás, tarde, temprano, pronto, bien, mal, apenas, adrede, despacio, recio, duro, fuerte, alto, más, menos, mucho, poco, casi, bastante, hartó, demasiado, tanto, muy, sólo, solo, solamente, sumamente, tremendamente, primeramente, posteriormente, cierto, cierta, ciertas, ciertamente, verdad, verdadero, verdadera, verdaderamente, indudablemente, también, seguramente, sí, no, tampoco, quizás, acaso, yo, tú, usted, él, ella, ello, nosotros, nosotras, vosotros, vosotras, ustedes, ellos, ellas, mío, mía, míos, mías, tuyo, tuya, tuyos, tuyas, suyo, suya, suyos, suyas, nuestro, nuestra, nuestros, nuestras, mi, su, sus, tu, este, éste, esté, esta, ésta, está, estamos, estáis, están, esto, estos, éstos, estas, éstas, eso, ese, ése, esa, ésa, esos, ésos, esas, éstas, aquel, aquella, aquellos, aquellas, quienes, cuya, cuyos, cuyas, quién, quiénes, qué, cuál, cuáles, cuándo, cuánto, cómo, alguien, nadie, algo, nada, alguno, algunos, alguna, algunas, cualquiera, quienquiera, todo, ninguno, ninguna, varios, ciertos, otro, mí, conmigo, ti, contigo, consigo, me, te, lo, la, le, se, nos, os, los, las, les, el, al, un, uno, una, unos, unas, algún, ser, es, soy, eres, somos, sois, estoy, atrás, por qué, estado, estaba, siendo, ambos, poder, puede, puedo, podemos, podéis, pueden, fui, fue, fuimos, fueron, hacer, hago, hace, hacemos, hacéis, hacen, cada, fin, incluso, primero, conseguir, consigue, consigues, conseguimos, consiguen, ir, voy, va, vamos, vais, van, vaya, bueno, ha, tener, tengo, tiene, tenemos, tenéis, tienen, saber, sabes, sabe, sabemos, sabéis, saben, último, largo, haces, muchos, intentar, intento, intenta, intentas, intentamos, intentáis, intentan, dos, usar, uso, usas, usa, usamos, usáis, usan,

emplear, empleo, empleas, emplean, empleamos, empleáis, valor, era, eras, éramos, eran, modo, podría, podrías, podríamos, podrían, podríais, del

Anexo 2. Medidas de comparación entre los algoritmos de clasificación.

Clasificador	Total accuracy	Macro-promedio Precision	Macro-promedio Recall	Macro-promedio F	Micro-promedio Precision	Micro-promedio Recall	Micro-promedio F
DynamicLM	0,797393455	0,799615973	0,78572294	0,78114343	0,797393455	0,797393455	0,797393455
LogisticRegression	0,795776256	0,801184232	0,779430118	0,775097147	0,795776256	0,795776256	0,795776256
Tfidf	0,746194825	0,731262193	0,736301199	0,719137898	0,746194825	0,746194825	0,746194825
NaiveBayes	0,70761035	0,752819182	0,677996859	0,687474745	0,70761035	0,70761035	0,70761035
Knn	0,423344749	0,639558532	0,414711165	0,460221883	0,423344749	0,423344749	0,423344749
Bernoulli	0,084208524	0	0,075315275	0	0,084208524	0,084208524	0,084208524

Anexo 3. Comparación entre herramientas de RI.

Características	Lemur	Indri	Weka	Lucene	LingPipe
Modelo de RI	Modelado de lenguaje, modelo Espacio-Vectorial, TF/IDF, Okapi e Inquiry.	Combinación del modelado de lenguaje y técnicas de red de inferencia.	-	Combinación del modelo Espacio Vectorial y el modelo puro booleano.	-
Tipo de fichero que es capaz de analizar gramaticalmente	Documentos TREC, Web TREC, texto plano, HTML, XML, PDF, documentos Word y PowerPoint (en Windows).	Texto plano, HTML, XML, PDF, documentos Word y PowerPoint (en Windows), cables de noticias TREC, documentos TREC y colecciones Web TREC.	-	Texto plano, XML, HTML, PDF, documentos Word.	Texto plano, HTML, XML.
Lenguaje de programación	C y C++	C y C++	Java	Java	Java
API Java	sí	sí	no	sí	sí
Tokenización	sí	sí	no	sí	sí
Eliminación de palabras vacías	no	no	no	sí	sí
Lematización	sí	sí	no	no	no
Segmentación	sí	sí	no	sí	sí
Clasificación	sí	no	sí	no	sí
Plataforma	Multiplataforma y la Web	Multiplataforma y la Web	Multiplataforma	Multiplataforma	Multiplataforma

Anexo 4. Diagrama de Gantt.

Tareas	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24	S25	S26
Análisis de Requisitos	x	x																									
Metodología de la Investigación	x	x	x	x	x																						
Fundamentación Teórica					x	x	x	x	x																		
Primer Corte de Tesis																											
Análisis y Diseño Preliminar			x	x	x	x	x	x	x	x																	
Diseño											x	x	x	x	x	x											
Implementación			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x				
Descripción de la Solución											x	x	x	x	x	x	x	x	x	x	x	x	x				
Segundo Corte de Tesis																											
Terminación del Documento de Tesis																								x			
Predefensa																											
Preparación de la Defensa																								x	x		
Defensa de la Tesis																											

Proyecto: Sistema de Clasificación Automática de Noticias a publicar en el periódico *jahora!* digital.

Autora: Yisel Clavel Quintero

Fecha: 04/01/2010

Semanas	Del	Al	Tareas
S0	04/01/2010	10/01/2010	
S1	11/01/2010	17/01/2010	
S2	18/01/2010	24/01/2010	
S3	25/01/2010	31/01/2010	
S4	01/02/2010	07/02/2010	
S5	08/02/2010	14/02/2010	
S6	15/02/2010	21/02/2010	
S7	22/02/2010	28/02/2010	
S8	01/03/2010	07/03/2010	Primer Corte
S9	08/03/2010	14/03/2010	Primer Corte
S10	15/03/2010	21/03/2010	
S11	22/03/2010	28/03/2010	
S12	29/03/2010	04/04/2010	
S13	05/04/2010	11/04/2010	
S14	12/04/2010	18/04/2010	
S15	19/04/2010	25/04/2010	
S16	26/04/2010	02/05/2010	Segundo Corte
S17	03/05/2010	09/05/2010	Segundo Corte
S18	10/05/2010	16/05/2010	
S19	17/05/2010	23/05/2010	
S20	24/05/2010	30/05/2010	
S21	31/05/2010	06/06/2010	Predefensa
S22	07/06/2010	13/06/2010	Predefensa
S23	14/06/2010	20/06/2010	
S24	21/06/2010	27/06/2010	
S25	28/06/2010	04/07/2010	Defensa
S26	05/07/2010	11/07/2010	Defensa

Anexo 5. Descripción de Casos de Uso.***Paquete Seguridad*****Caso de Uso: Iniciar Sesión**

Curso Básico: El Editor Web entra sus datos (nombre de usuario y contraseña) en la Interfaz de Autenticación del sistema y presiona el botón Aceptar. El sistema verifica que el usuario exista y que la contraseña sea correcta, inicia la sesión y visualiza la Interfaz Principal.

Cursos Alternos:

El usuario no existe: El sistema muestra un mensaje de error que dice que los datos son incorrectos. El Editor Web acepta el mensaje y el sistema muestra la Interfaz de Autenticación.

Contraseña incorrecta: El sistema muestra un mensaje de error que dice que los datos son incorrectos. El Editor Web acepta el mensaje y el sistema muestra la Interfaz de Autenticación.

Caso de Uso: Cerrar Sesión

Curso Básico: El Editor Web presiona el menú Archivo opción Cerrar Sesión en la Interfaz Principal. El sistema muestra un mensaje de confirmación. El usuario presiona el botón Aceptar y el sistema cierra la sesión y muestra la Interfaz de Autenticación.

Curso Alterno:

El Editor Web presiona Cancelar: El sistema retorna a la Interfaz Principal.

Paquete Clasificación**Caso de Uso: Visualizar Noticias**

Curso Básico: El Editor Web da doble clic sobre una noticia o la selecciona y presiona la tecla *Enter* en la Interfaz Principal, y el sistema muestra en otra ventana el contenido de la noticia seleccionada.

Curso Alterno:

Error al leer archivo: El sistema muestra un mensaje de error. El Editor Web acepta el mensaje y el sistema muestra la Interfaz Principal.

Caso de Uso: Clasificar Noticias Internas

Curso Básico: El Editor Web da doble clic o *Enter* en el sistema para abrir el sistema. El sistema clasifica las noticias extraídas de la BD y muestra los resultados en la pestaña Noticias Internas de la Interfaz Principal.

Curso Alterno:

Error de Conexión: El sistema muestra un mensaje de error. El Editor Web da clic en Aceptar y el sistema muestra la Interfaz Principal.

Caso de Uso: Cargar Noticias

Curso Básico: El Editor Web presiona el botón Agregar en la Interfaz Principal y el sistema muestra una ventana de selección de archivos. El Editor Web selecciona uno o varios archivos y presiona el botón Aceptar o la tecla *Enter* del teclado, y el sistema muestra una lista de los archivos seleccionados en la pestaña Noticias de la Interfaz Principal.

Curso Alterno:

Cancelar selección de archivos: El sistema cierra la ventana de selección de archivos y muestra la Interfaz Principal.

Caso de Uso: Escoger Noticias

Curso Básico: El Editor Web selecciona uno o varios archivos de la lista de archivos en la Interfaz Principal y el sistema activa el botón Clasificar.

Caso de Uso: Clasificar Noticias

Curso Básico: El Editor Web presiona el botón Clasificar en la Interfaz Principal y el sistema clasifica las noticias seleccionadas y muestra un mensaje diciendo que la clasificación se terminó correctamente. El usuario acepta el mensaje y el sistema muestra el resultado de la clasificación y activa el botón Guardar.

Curso Alterno:

Error en la clasificación: El sistema muestra un mensaje de error. El usuario acepta el mensaje y el sistema muestra la Interfaz Principal.

Caso de Uso: Guardar Clasificación NI

Curso Básico: El Editor Web selecciona las categorías que aprueba, una por cada noticia; o no selecciona ninguna si está de acuerdo con el resultado; luego presiona el botón Guardar en la pestaña Noticias Internas de la Interfaz Principal. El sistema guarda en la BD del periódico digital el resultado de la clasificación en las categorías seleccionadas.

Curso Alternativo:

Error de conexión: El sistema muestra un mensaje de error. El Editor Web acepta el mensaje y el sistema muestra la Interfaz Principal.

Caso de Uso: Guardar Clasificación N

Curso Básico: El Editor Web selecciona las categorías que aprueba, una por cada noticia; o no selecciona ninguna si está de acuerdo con el resultado; luego presiona el botón Guardar en la pestaña Noticias de la Interfaz Principal. El sistema guarda en la BD del periódico digital el resultado de la clasificación en las categorías seleccionadas.

Curso Alternativo:

Error de conexión: El sistema muestra un mensaje de error. El Editor Web acepta el mensaje y el sistema muestra la Interfaz Principal.

Paquete Servicio**Caso de Uso: Actualizar Conjunto de Entrenamiento**

Curso Básico: El Reloj marca la hora predefinida por el usuario y el sistema actualiza el conjunto de entrenamiento con las noticias de la BD.

Curso Alternativo:

Error de conexión: El sistema muestra un mensaje de error. El Editor Web acepta el mensaje.

Caso de Uso: Actualizar ahora

Curso Básico: El Editor Web presiona el botón Actualizar ahora en la Interfaz de Administración de Servicio y el sistema actualiza el conjunto de entrenamiento con noticias extraídas de la BD.

Curso Alterno:

Error de conexión: El sistema muestra un mensaje de error. El Editor Web acepta el mensaje. El sistema muestra la Interfaz de Administración de Servicio.

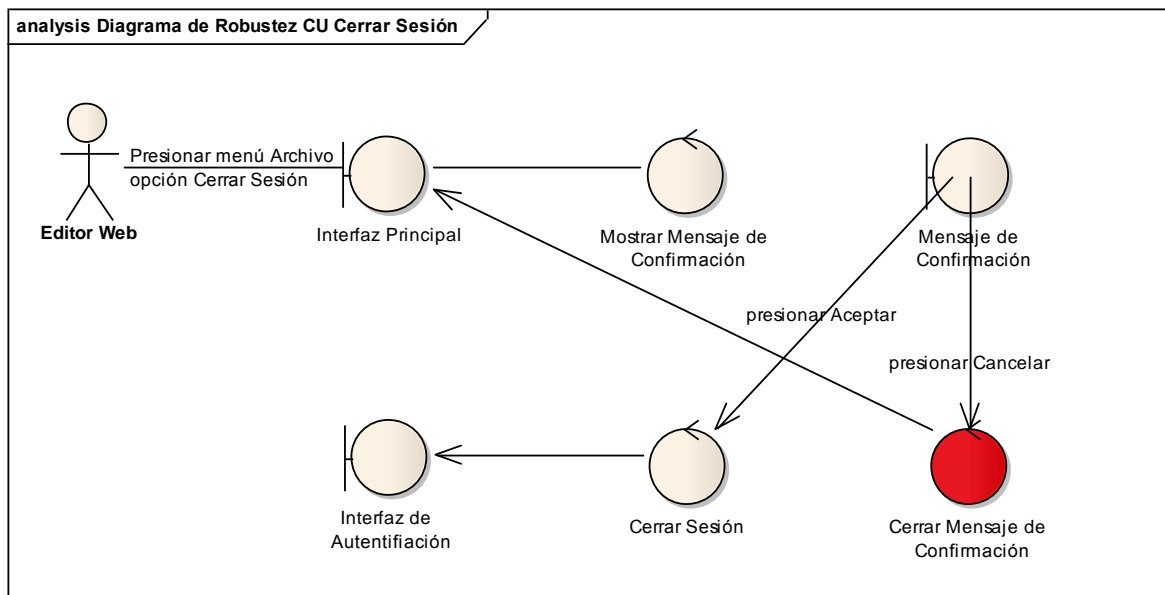
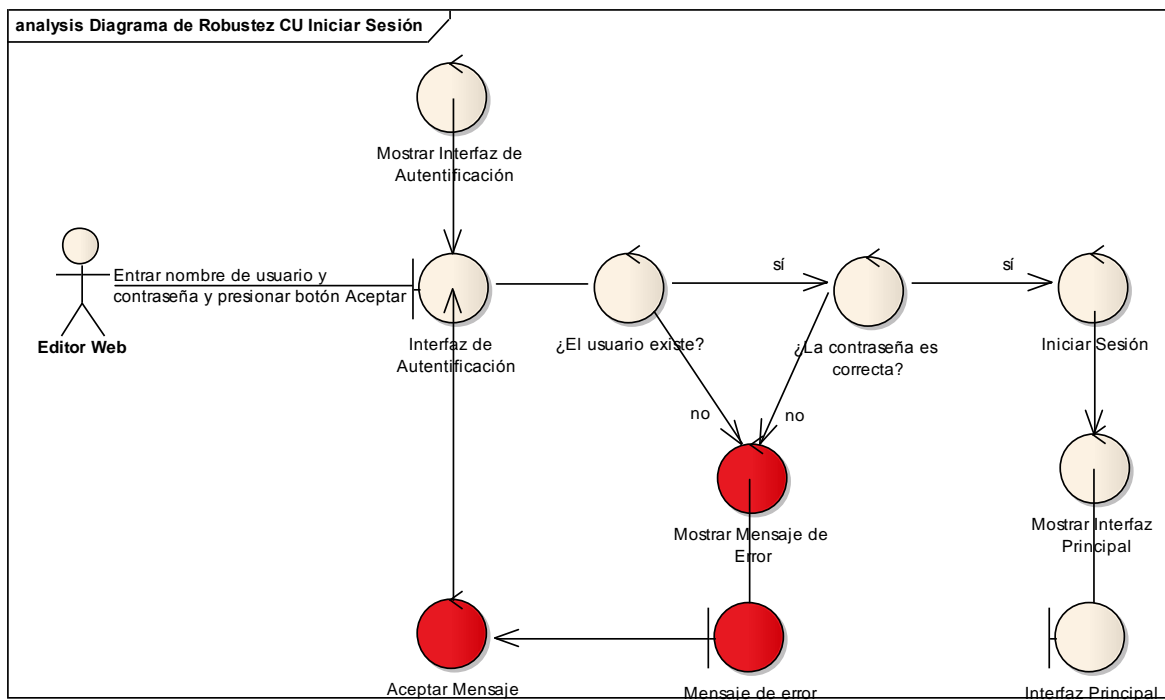
Caso de Uso: Especificar Horario de Actualización

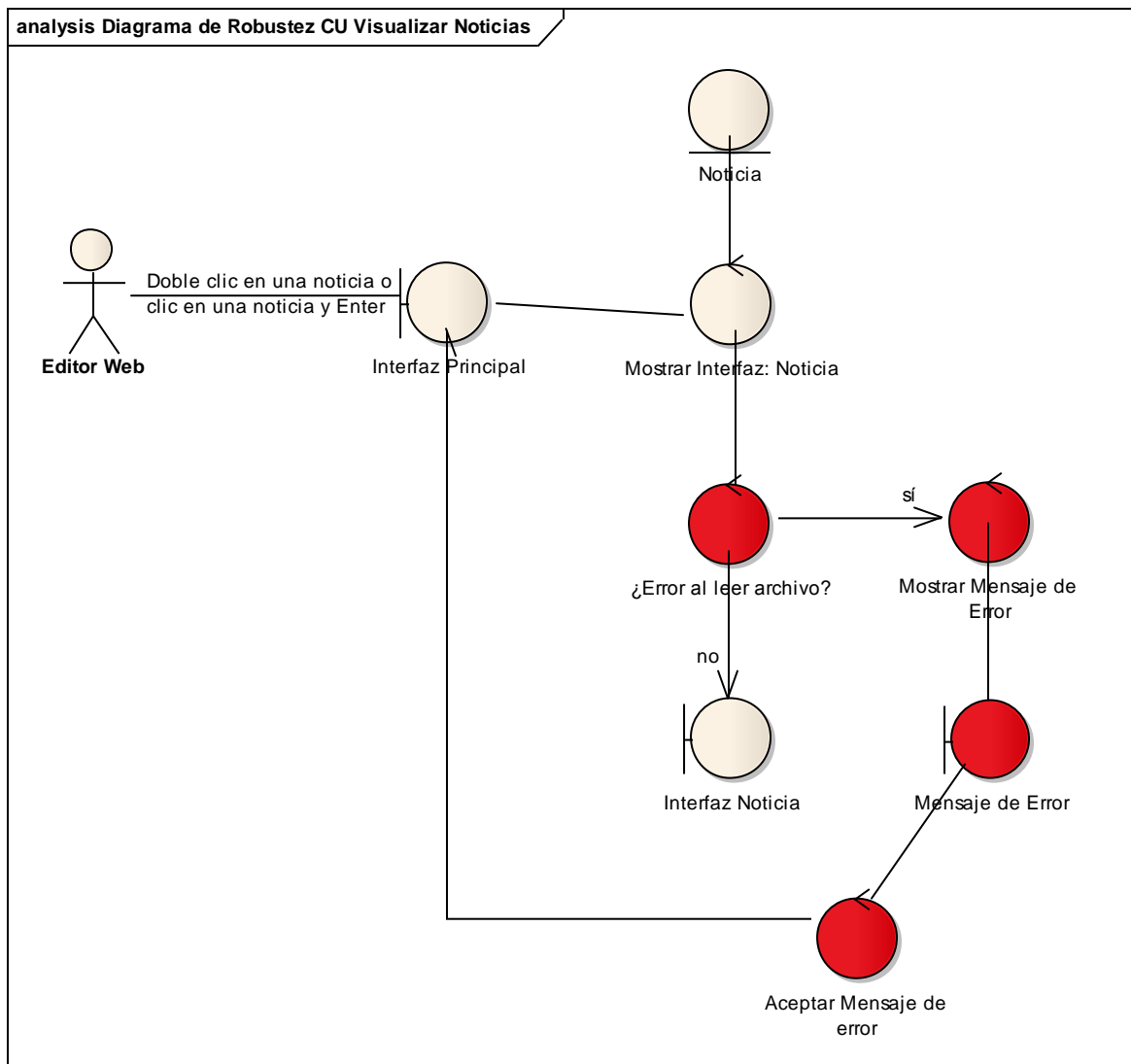
Curso Básico: El Editor Web especifica el horario de actualización del conjunto de entrenamiento y presiona el botón Guardar. El sistema guarda el horario definido.

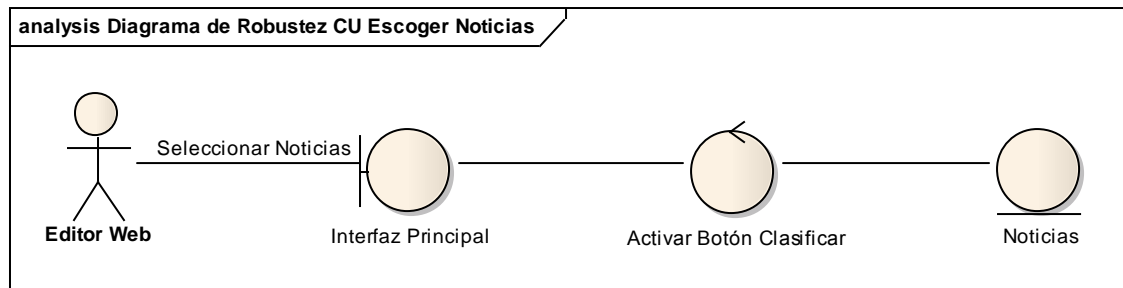
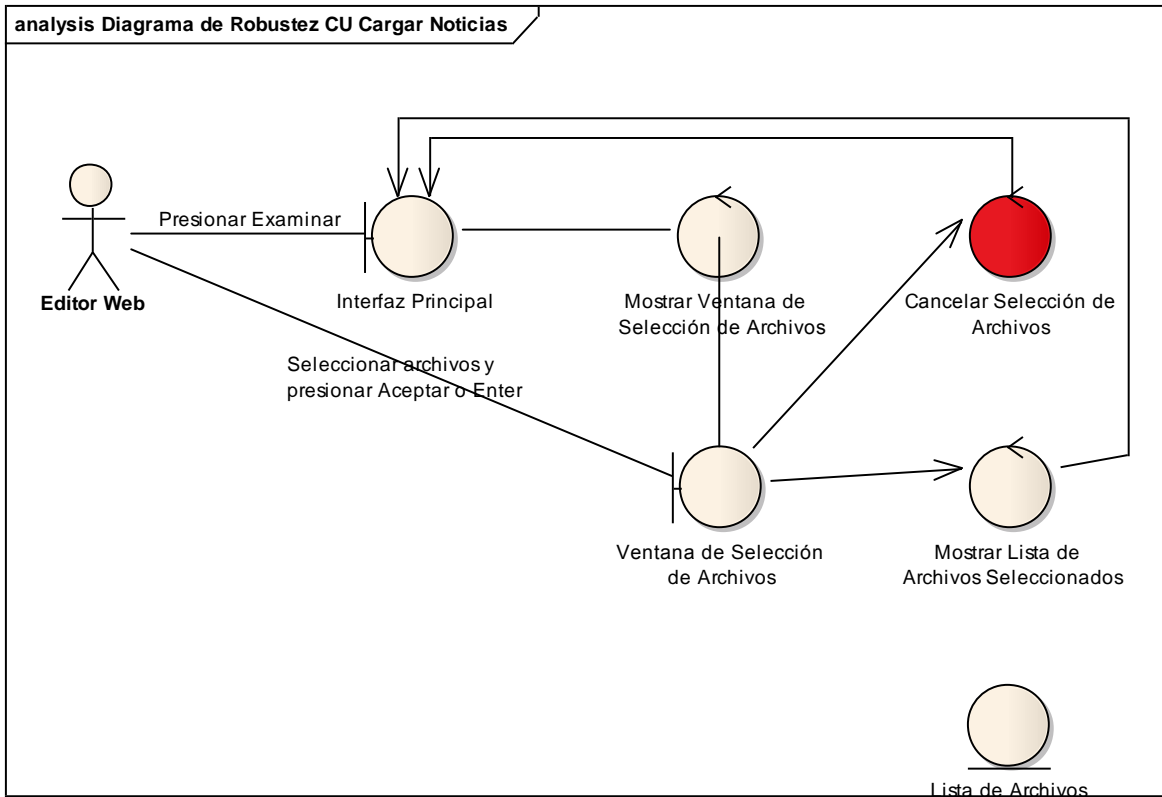
Curso Alterno:

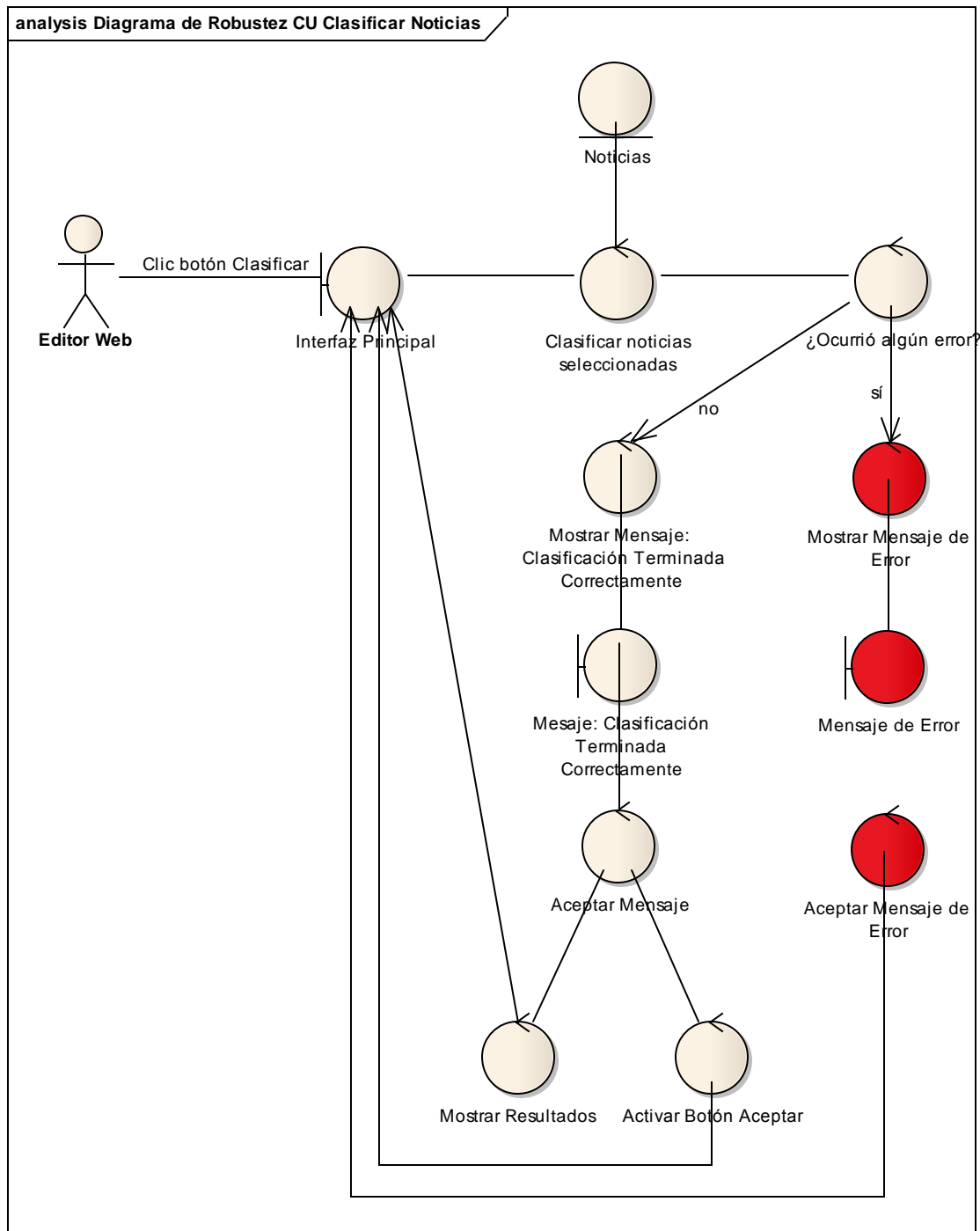
Horario incorrecto: El sistema muestra un mensaje de error. El Editor Web acepta el mensaje y el sistema muestra la Interfaz de Administración de Servicio.

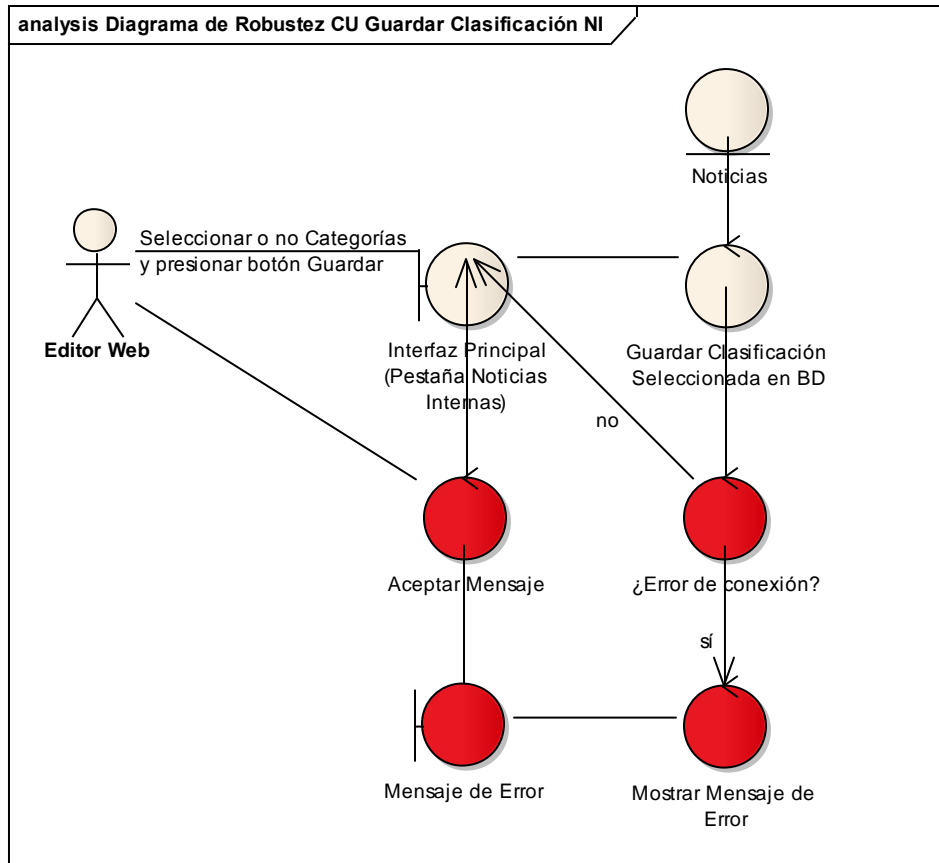
Anexo 6. Diagrama de Robustez del Paquete Seguridad.

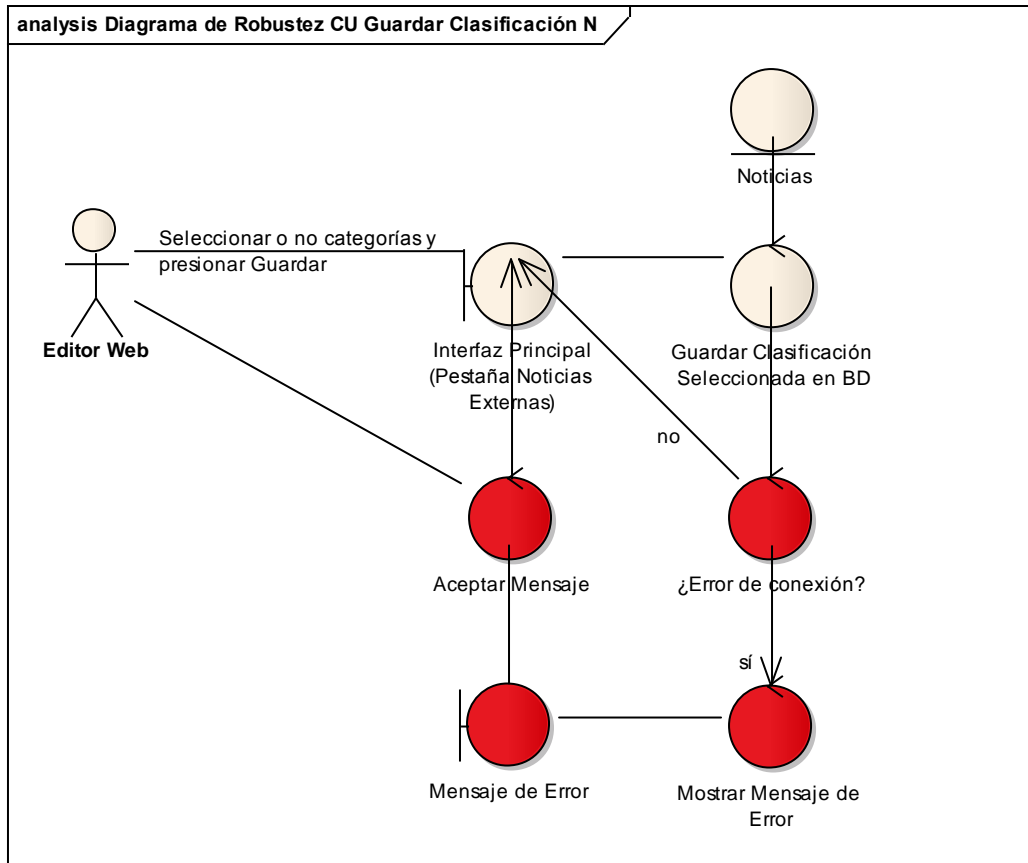


Anexo 7. Diagramas de Robustez del Paquete Clasificación.

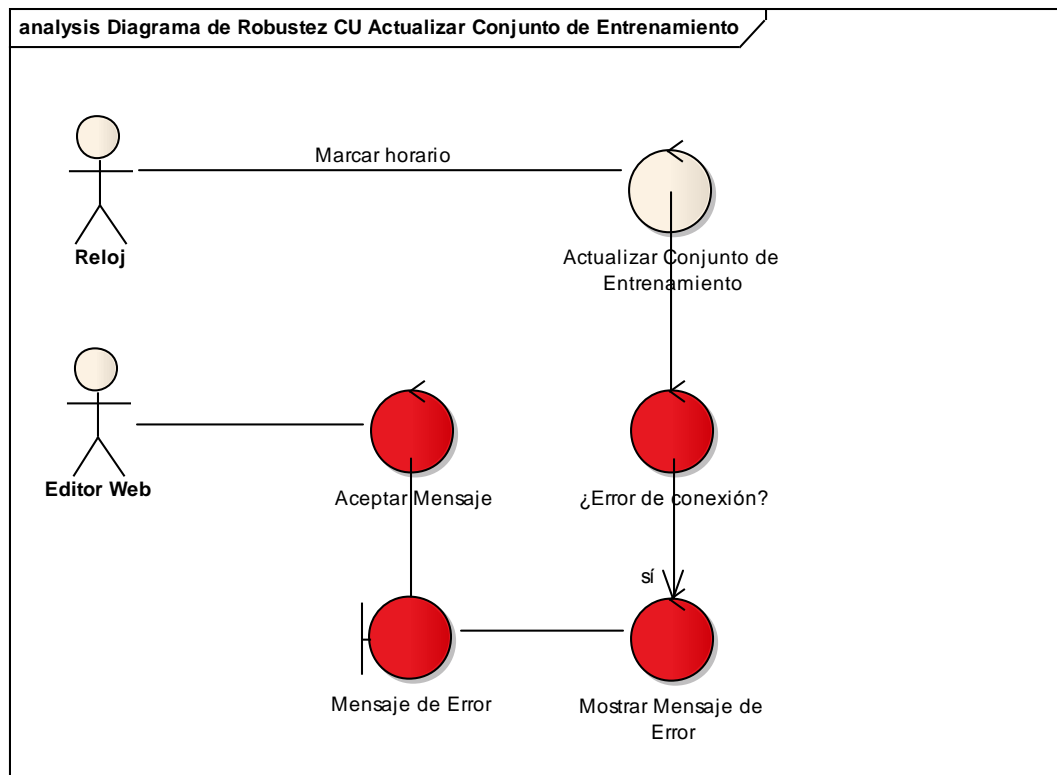


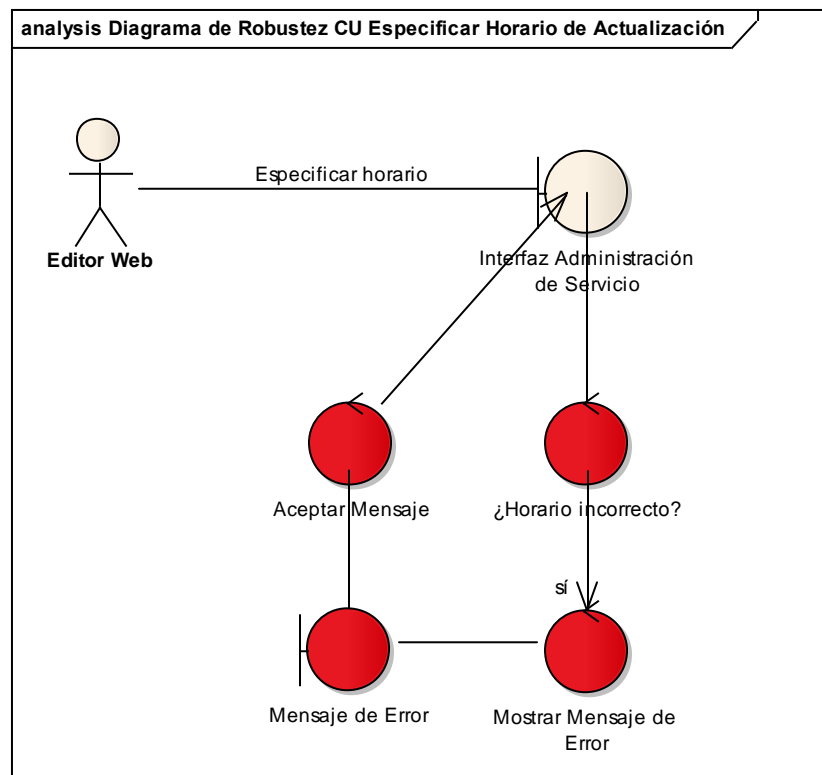
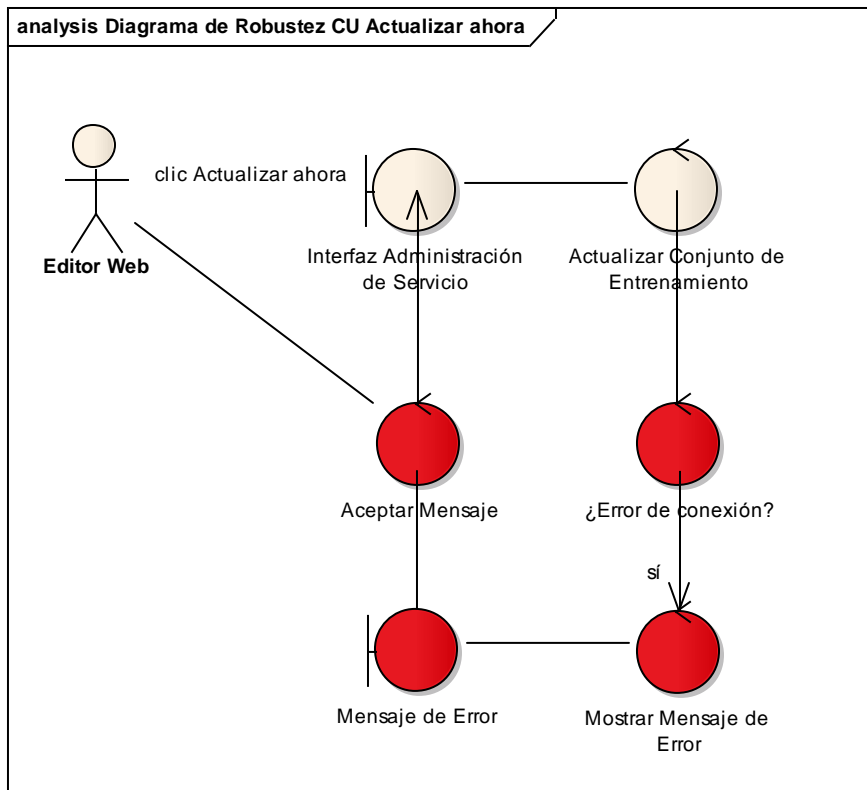




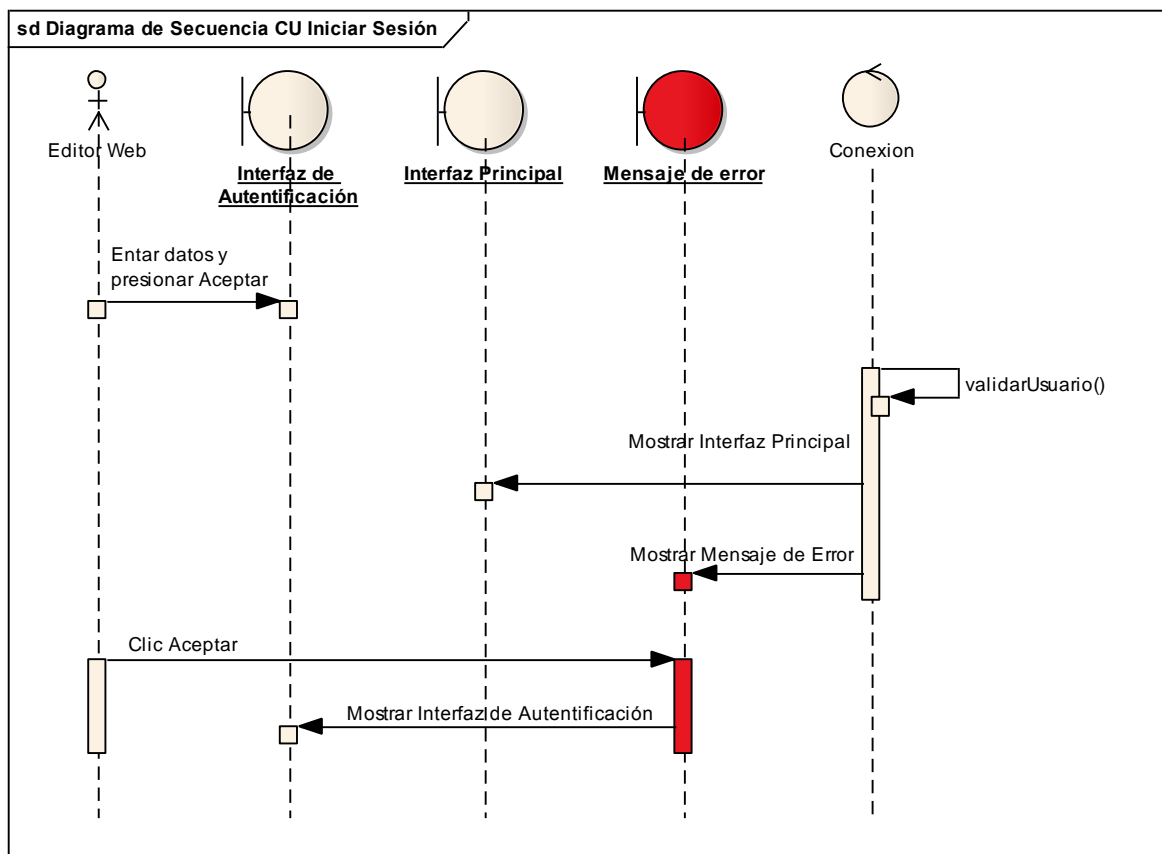


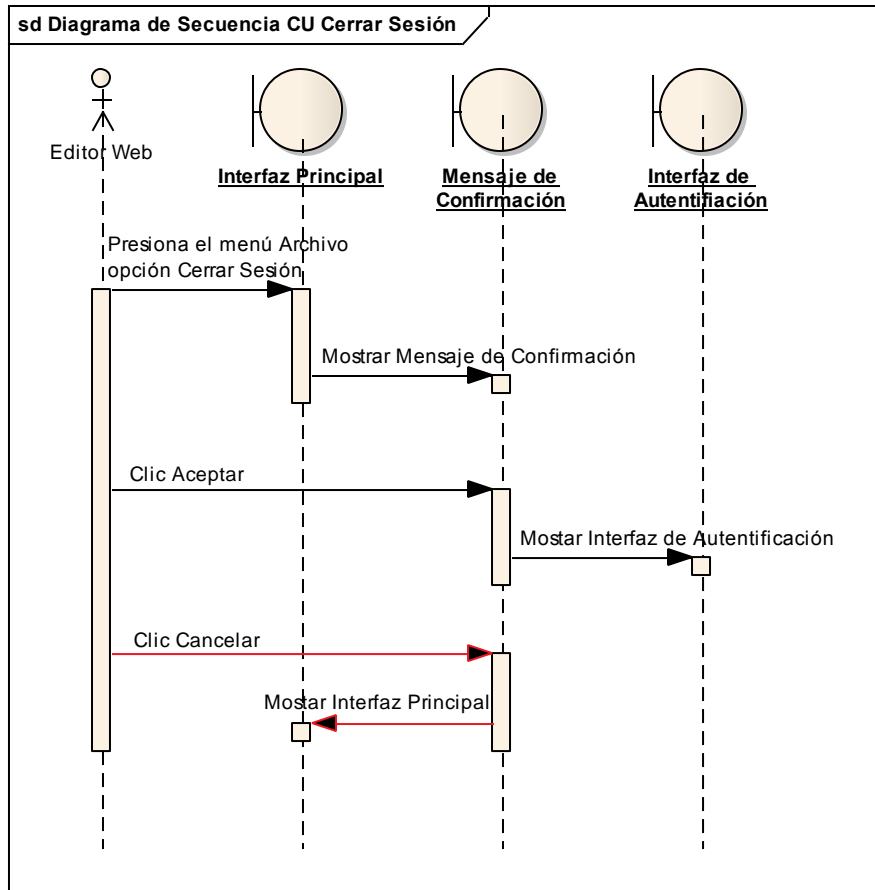
Anexo 8. Diagramas de Robustez del Paquete Servicio.



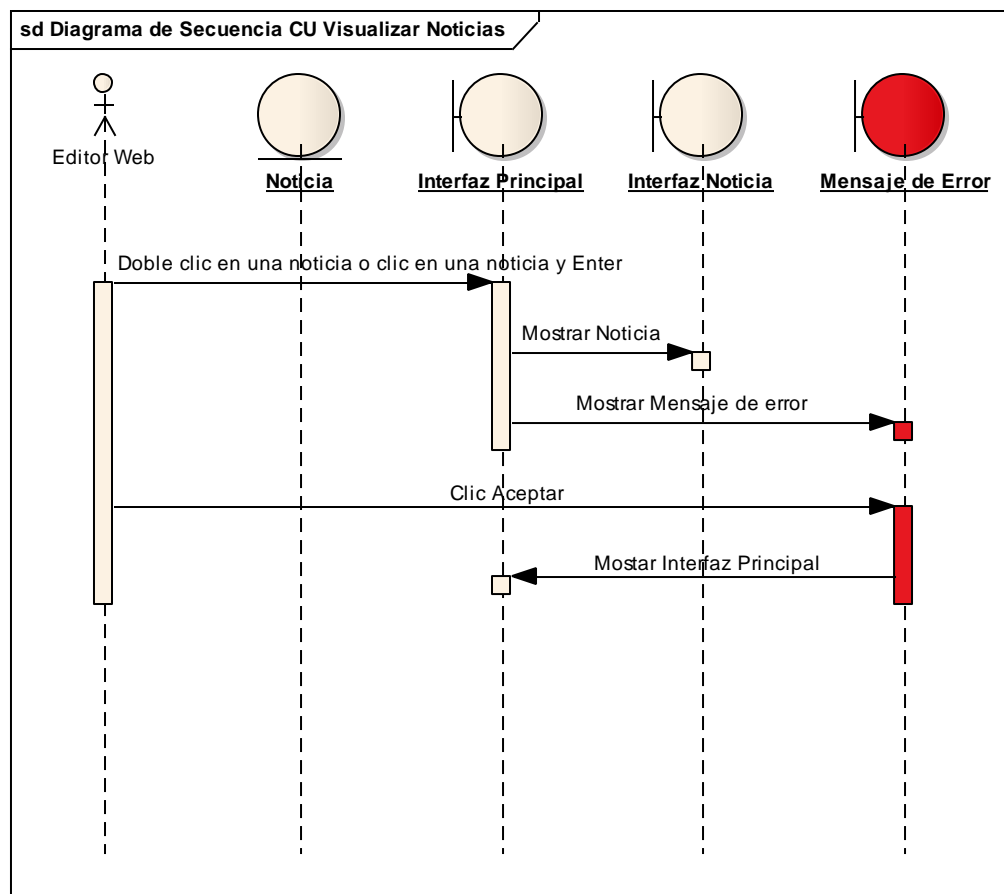


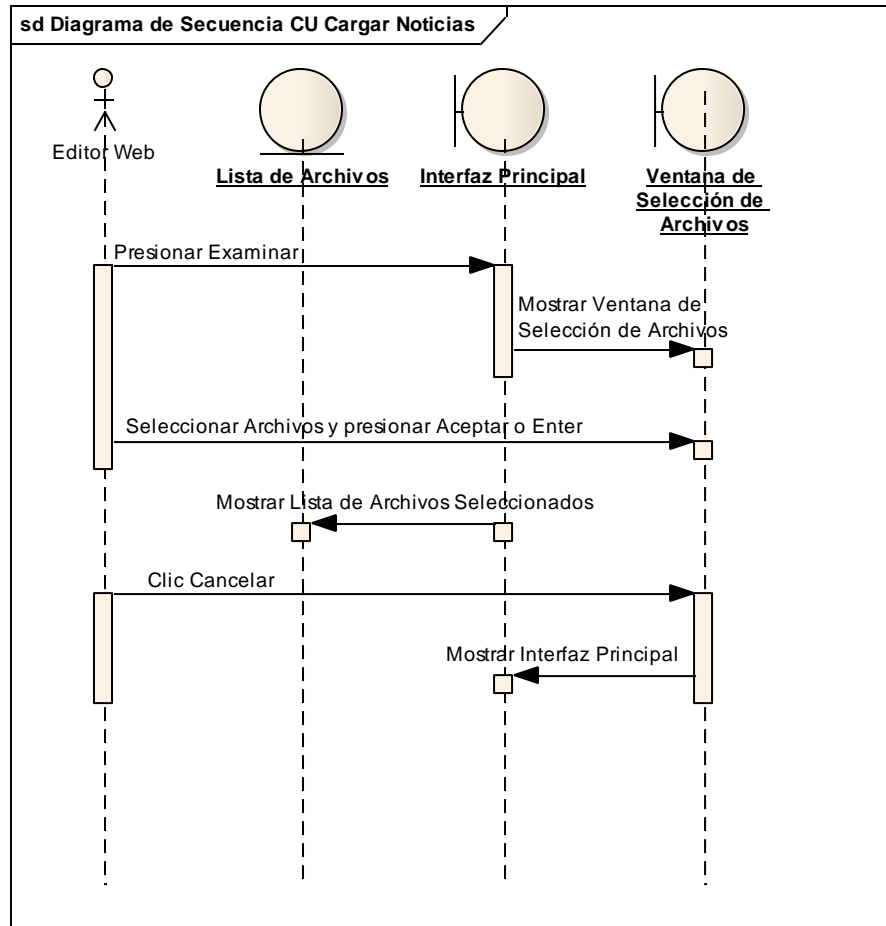
Anexo 9. Diagramas de Secuencia del Paquete Seguridad.

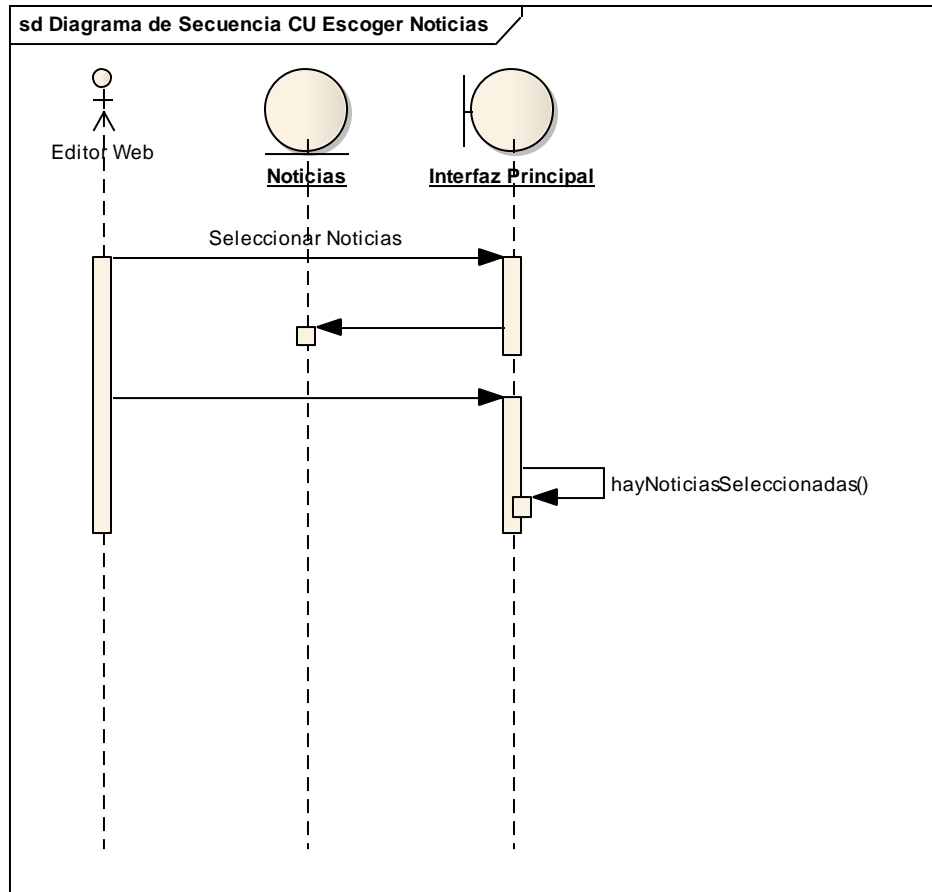


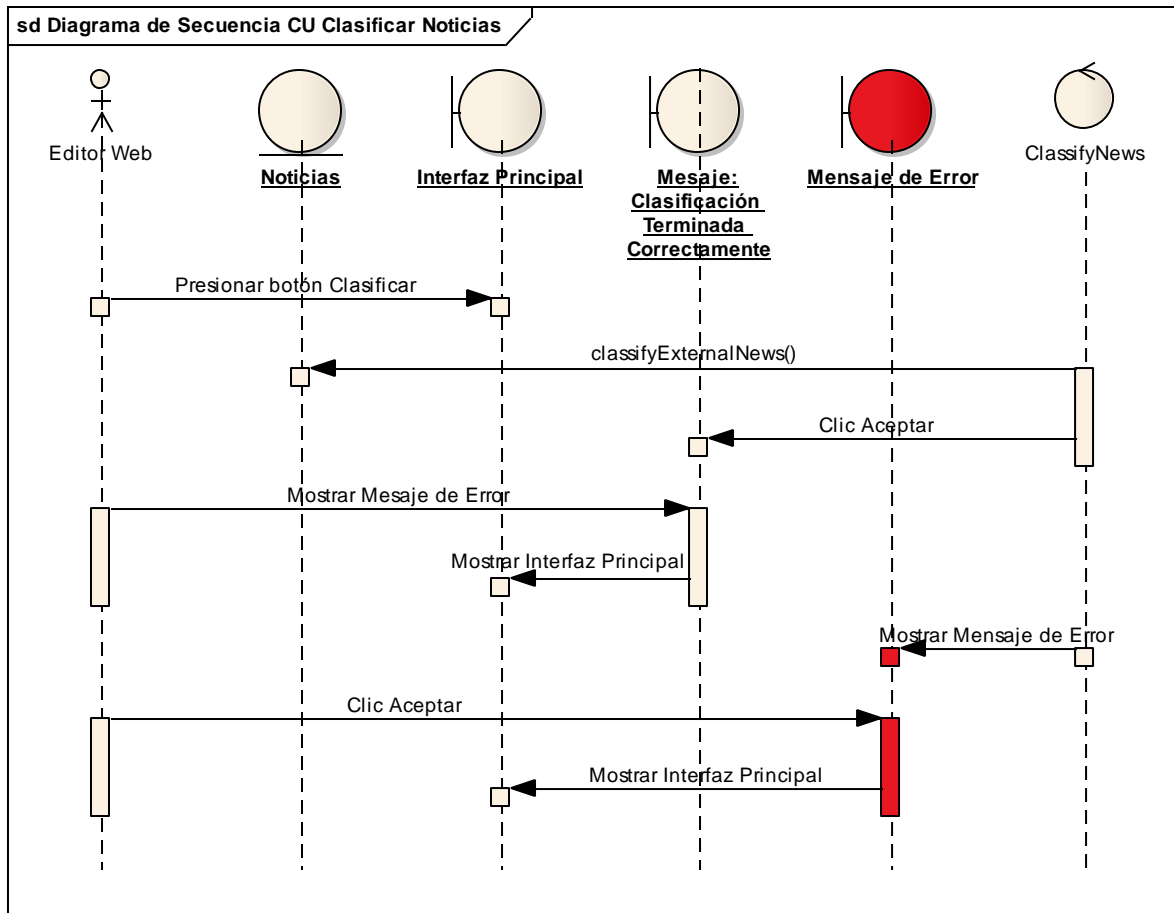


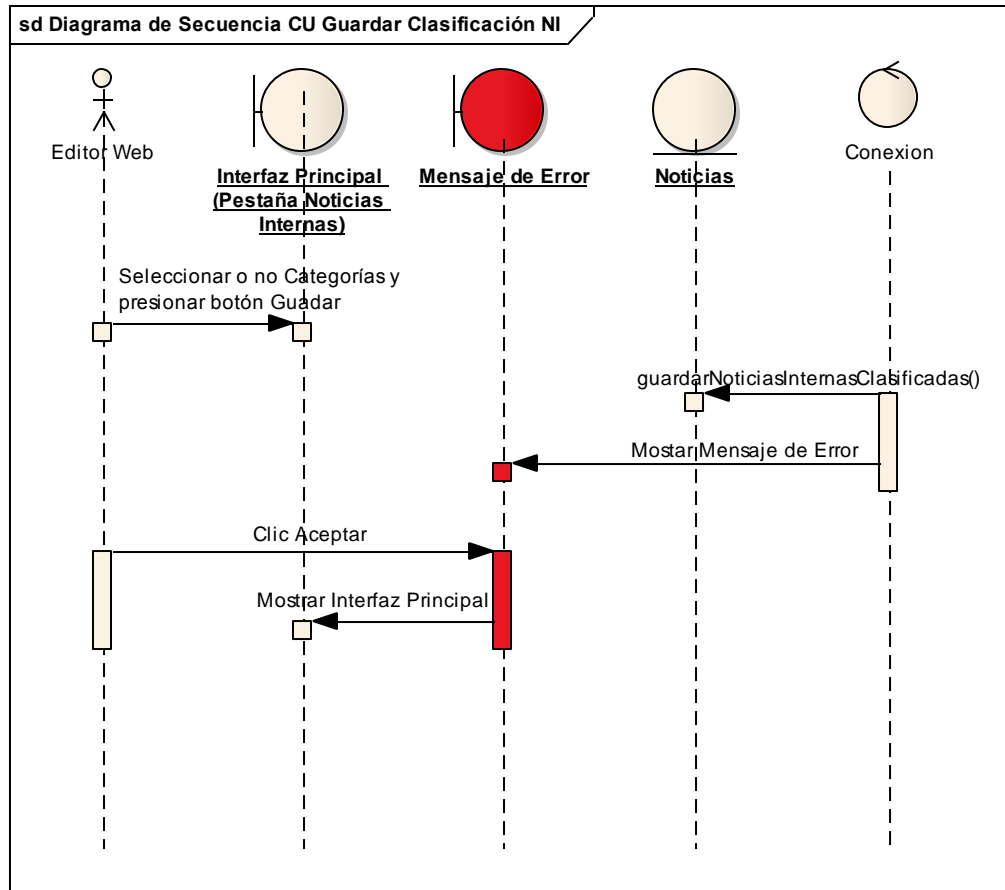
Anexo 10. Diagramas de Secuencia del Paquete Clasificación.

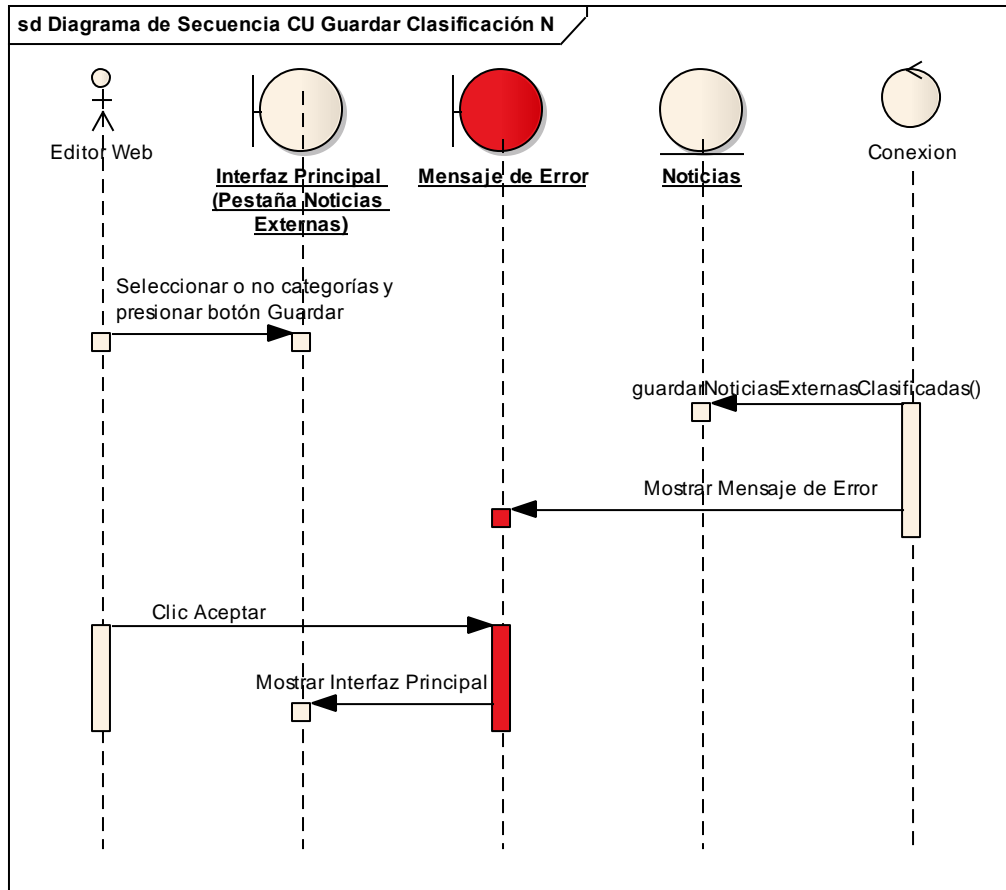




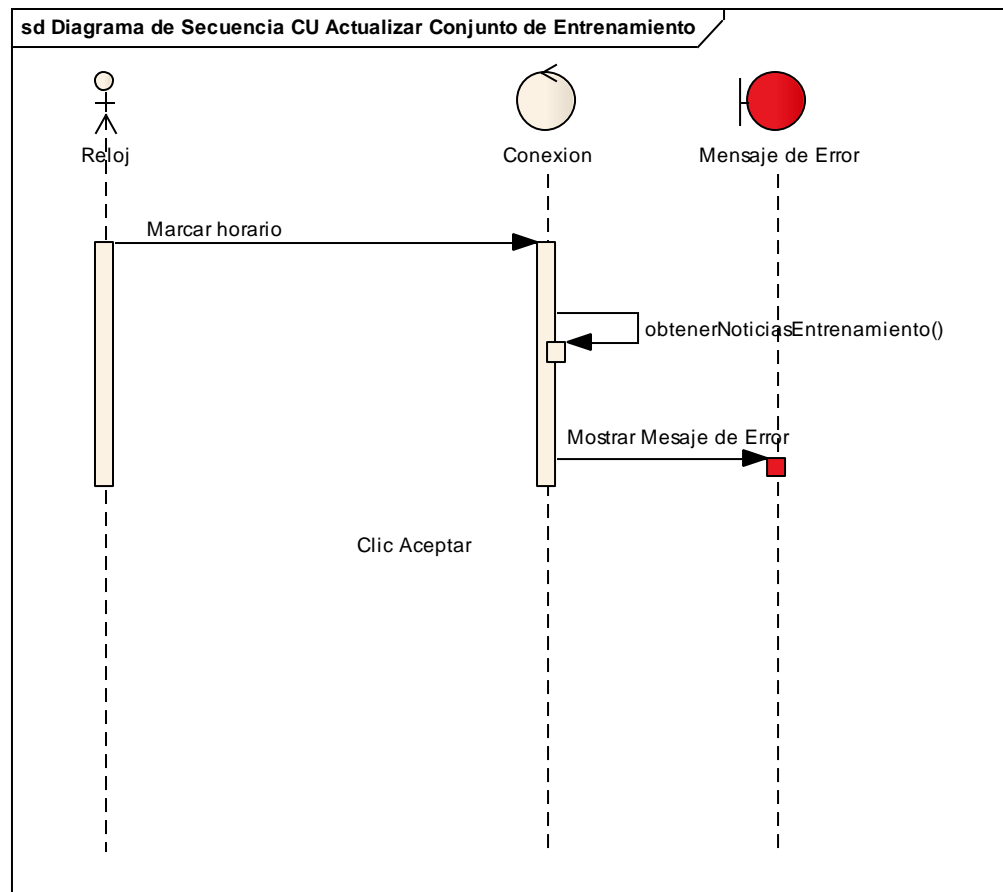


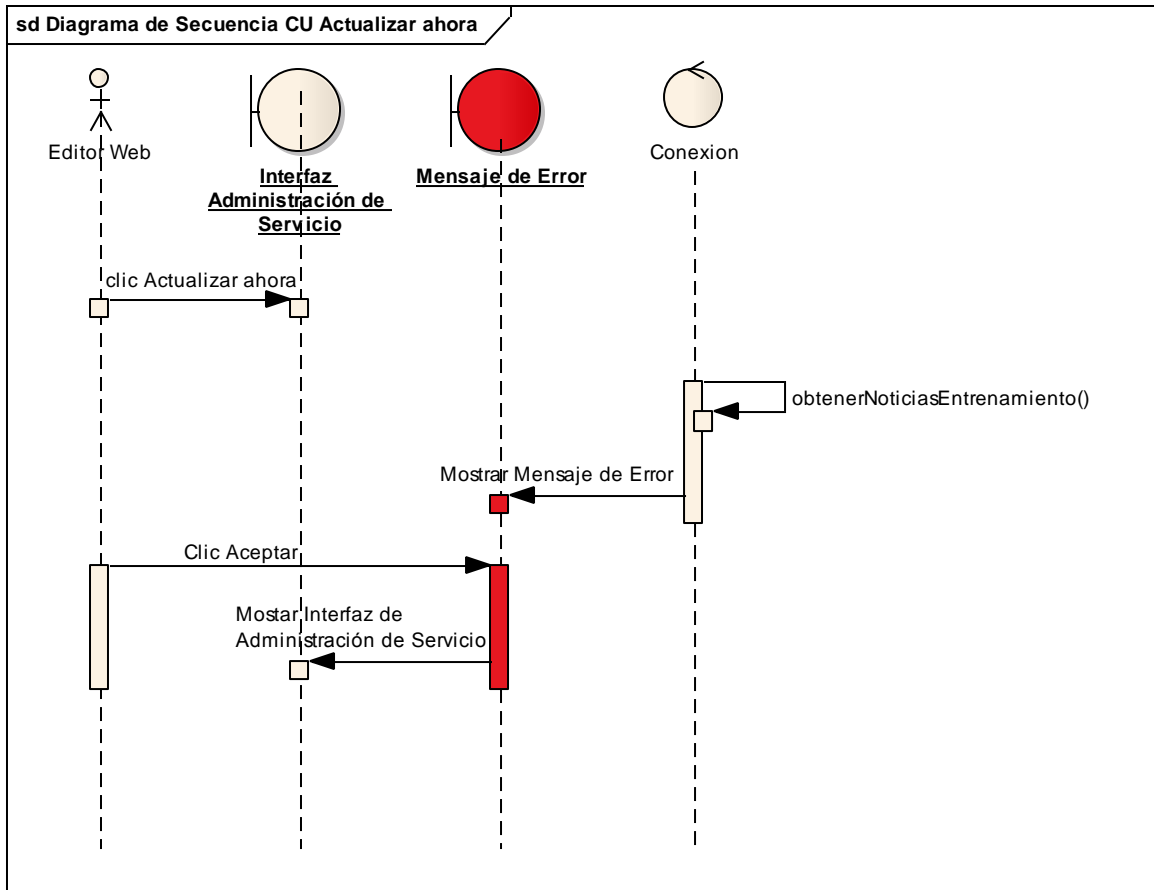


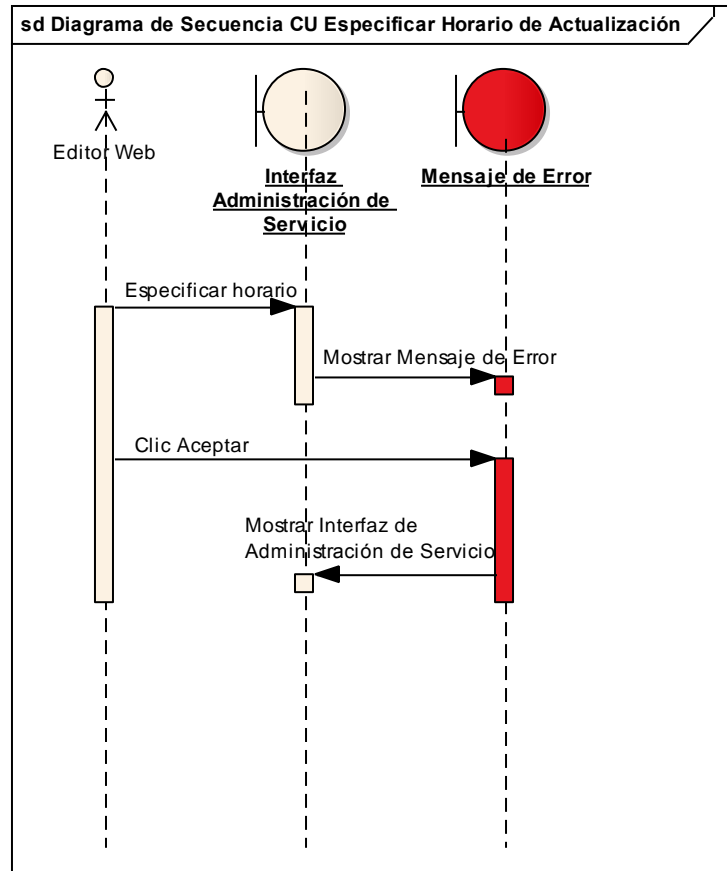




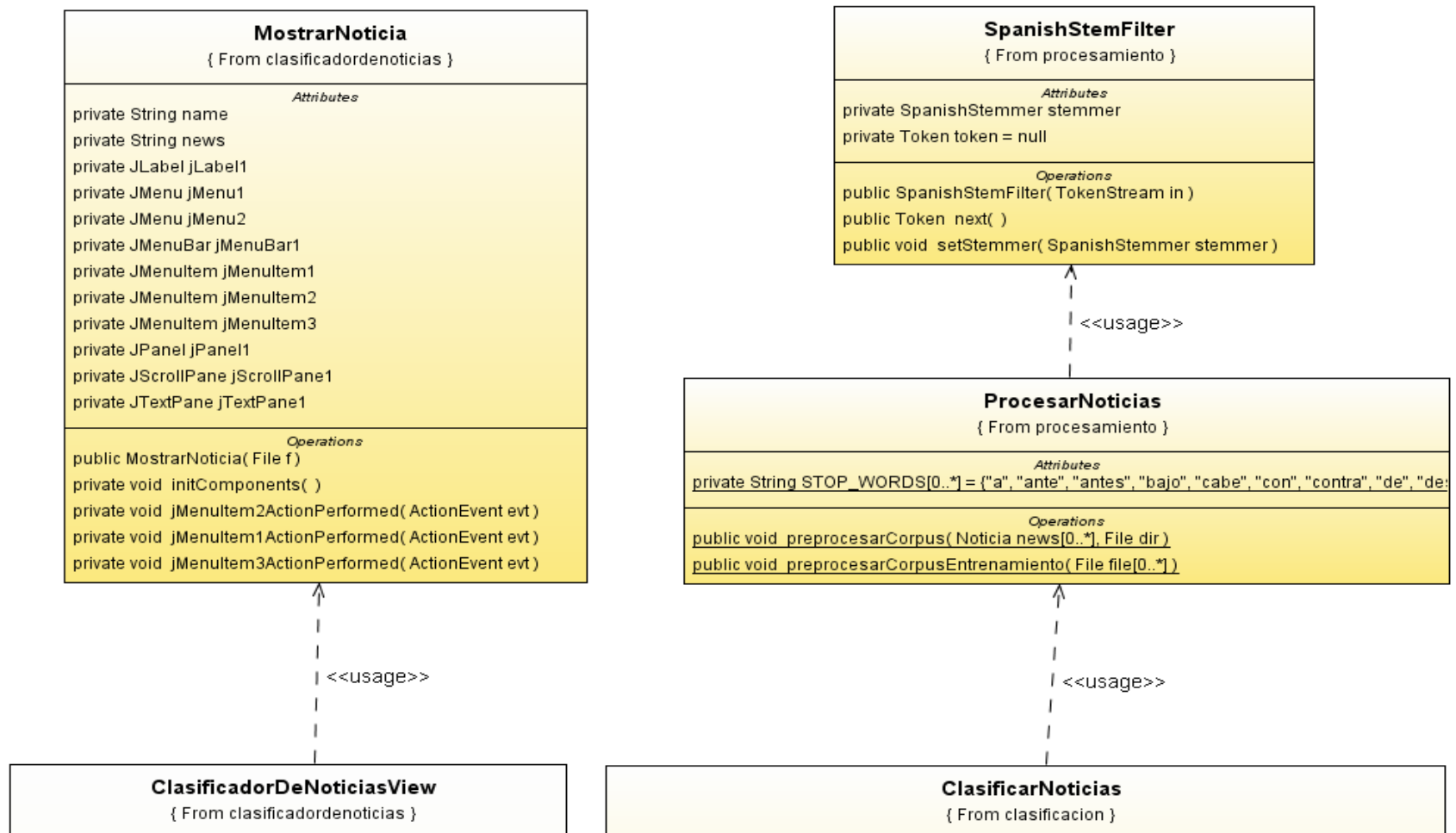
Anexo 11. Diagramas de Secuencia del Paquete Servicio.

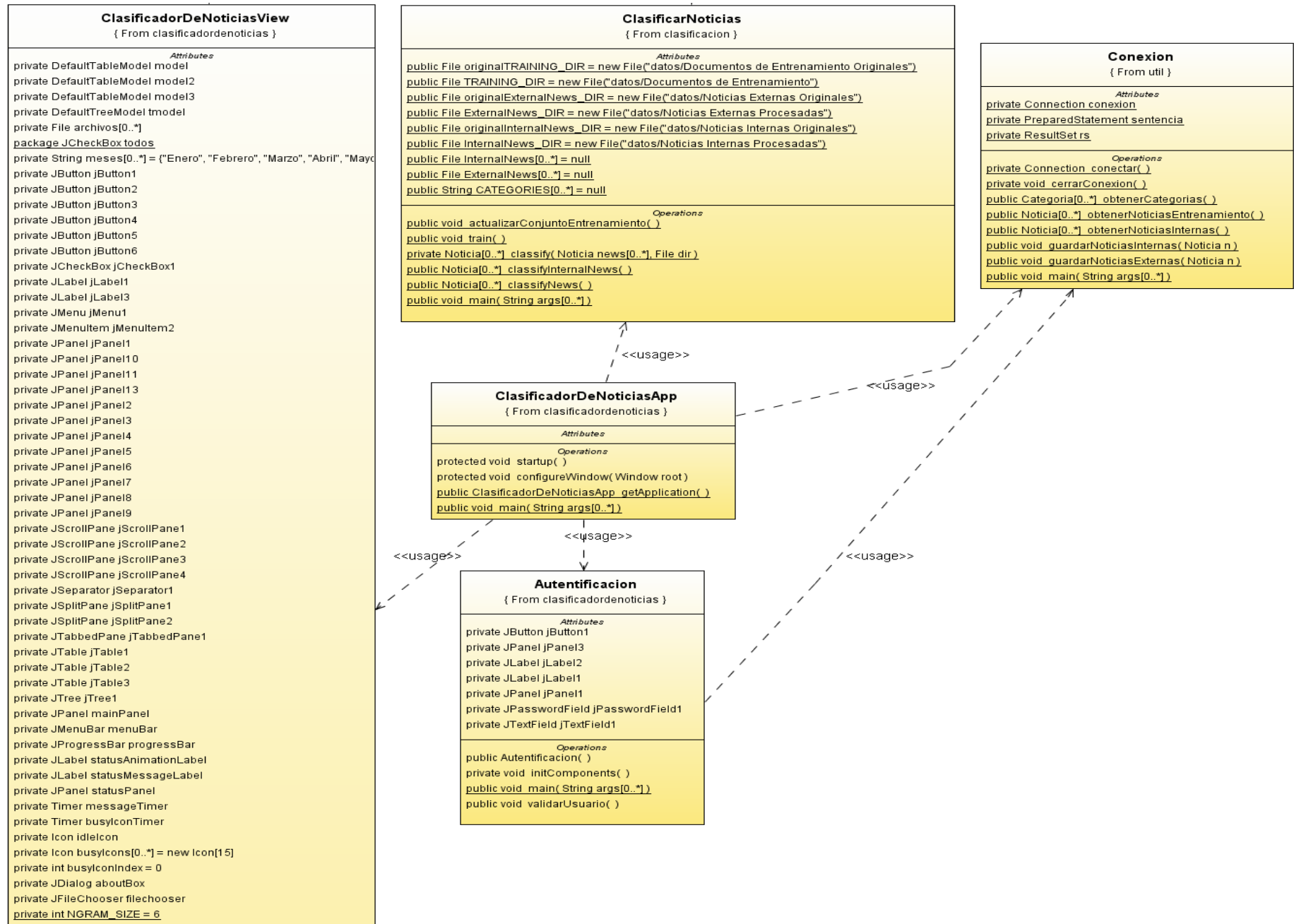


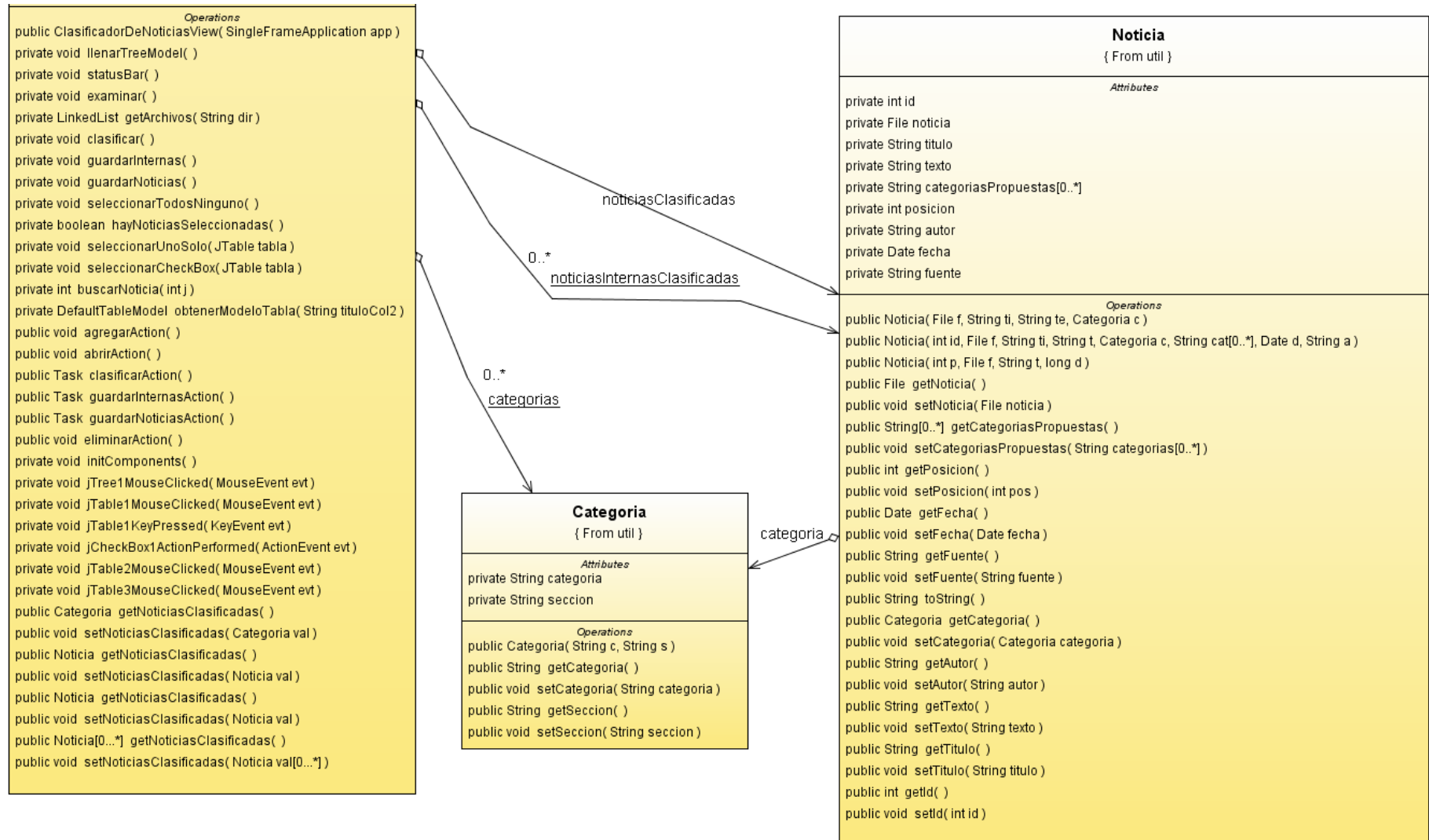


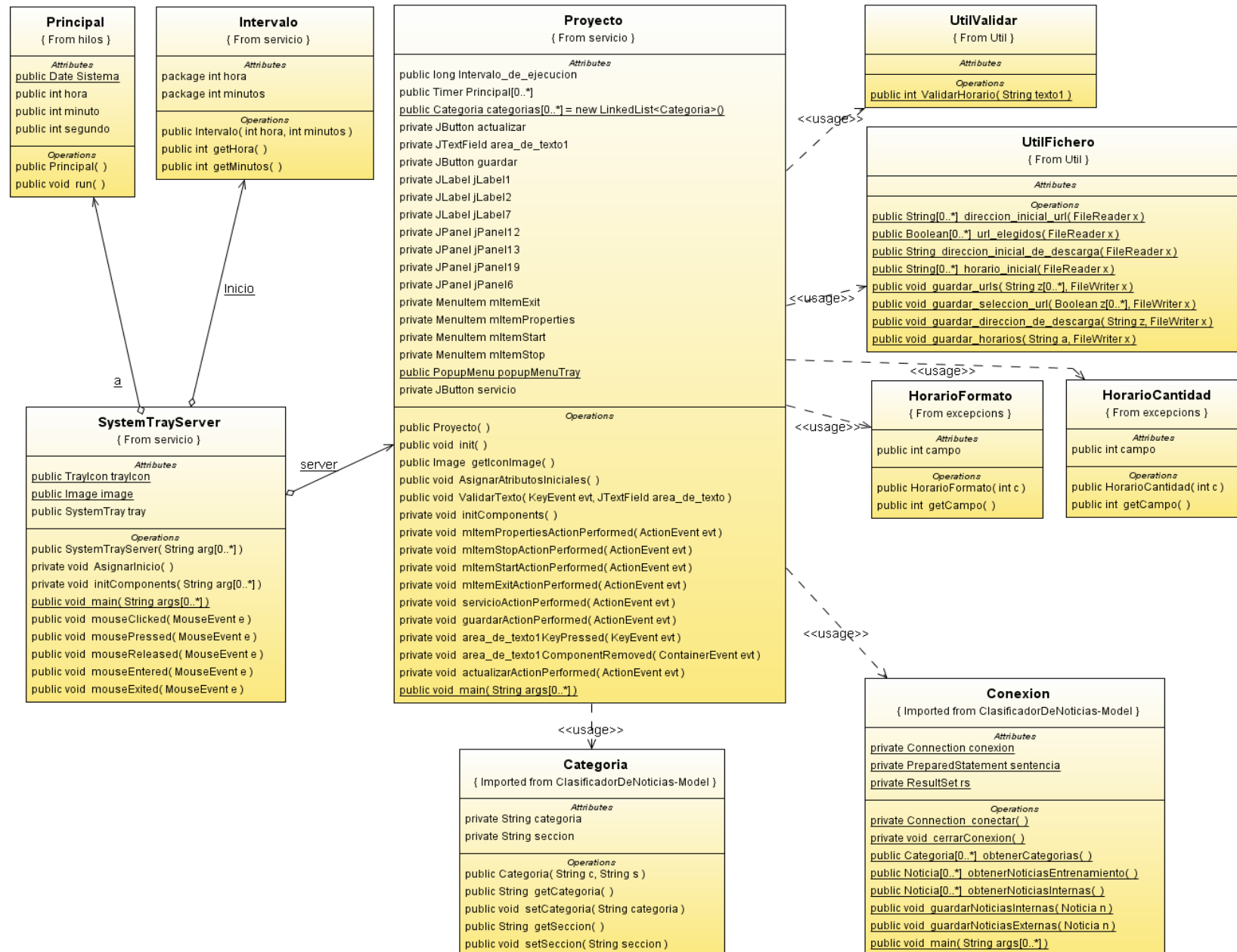


Anexo 12. Diagrama de Clases.









Anexo 13. Estándar de código utilizado.

Historial de versiones del documento:

Fecha	Versión	Descripción	Autor
14/11/2009	1.0	Sistema de Clasificación Automática de Noticias a publicar en el periódico <i>¡ahora!</i> digital	Yisel Clavel Quintero

1 Organización del código**1.1 Aspectos generales**

- Idioma:

Los idiomas que se van a emplear para nombrar los distintos elementos son el Inglés y el Español.

- Identación:

Elemento	Valor
clases	0
métodos	1
atributos	1

- Anidamiento:

El anidamiento va a ser absoluto para cualquier tipo de instrucción a no más de 5 niveles (es).

- Tamaño máximo de líneas:

Las líneas de código no deben exceder los 80 caracteres. Por su parte, las de ruptura (continuación de una línea de código que excedió los 80 caracteres) no deben exceder a 35 caracteres.

- Módulos:

Los módulos no deben contener más de 10000 líneas de código.

- Apertura y cierre de ámbito:

Los ámbitos van a ser abiertos y cerrados en la misma línea de la sentencia que los precede.

1.2 Líneas y espacios en blanco

Líneas en blanco:

Se deben usar líneas en blanco antes y después de:

- La declaración de una estructura o una clase.
- La implementación del método de una clase.
- Comentarios no relacionados con el código.
- Bloques de códigos complejos.

Espacios en blanco:

Se deben usar espacios en blanco antes y después de:

- Operadores lógicos.
- Operadores matemáticos.

1.3 Organización del código (Ficheros)

La carpeta raíz del proyecto se va a llamar GrafoGS. El sistema va a ser único, lo cual significa que no va a estar subdividido en subsistemas.

Todos los ficheros van a ser almacenados en la carpeta raíz sin importar tipo.

2 Comentarios

Los comentarios serán escritos en Inglés.

2.1 Comentarios de una línea

Se deben usar este tipo de comentarios en los siguientes casos:

- Antes de la declaración de los métodos.
- Al final de la línea que necesite aclaración.

2.2 Comentarios en bloque

Estos comentarios se aplicarán en los casos que se listan a continuación, y con la estructura mostrada:

Inicio de cada clase

Estructura:

```
/* para lo que sirve la clase.....
```

```
.....
```

```
*/
```

Antes de la declaración de los métodos si el comentario excede una línea

Estructura:

/* comentario.....

.....

.....

*/

3 Nombres

3.1 Nombre para Paquetes

Los Paquetes no requieren prefijo.

Las letras de la palabra estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.2 Nombre para Unidades

Las Unidades no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.3 Nombre para Interfaces

Las Interfaces no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.4 Nombre para Clases

Las Clases no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.5 Nombre para Objetos

Los Objetos no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, y la longitud de este no debe exceder los 15 caracteres.

3.6 Nombre para Atributos pasivos

Los Atributos pasivos no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.7 Nombre para Métodos

Los Métodos no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.8 Nombre para Constructores

Los Constructores no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.9 Nombre para Parámetros

Los Parámetros no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.10 Nombre para Excepciones y sus objetos

Las Excepciones y sus objetos no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, y la longitud de este no debe exceder los 20 caracteres.

3.11 Nombre para Constantes

Las Constantes no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.12 Nombre para Arreglos

Los Arreglos no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.13 Nombre para Variables locales

Las Variables locales no requieren prefijo.

Las letras de la palabra estarán en Notación Camell. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

3.14 Nombre para Contadores de ciclo

Se van a utilizar letras.

4 Declaraciones e inicializaciones

4.1 Aspectos generales

4.1.1 Tipos de datos

No va a importar el lugar donde sean declarados los tipos de datos

4.2 Declaración de clases

4.2.1 Orden de declaración

Por niveles de visibilidad

Los niveles de visibilidad van a ser declarados en orden ascendente, por lo que, primero se declarará *private*, luego *protected*, y por último *public*).

Dentro de un nivel de visibilidad

Deben declarar primero los atributos y luego los métodos.

En grupos de elementos

Deben ser declarados sin importar el orden.

4.2.2 Constructores y destructores

Los constructores y destructores serán declarados encabezando el segmento de los métodos.

4.2.3 Funciones sobrecargadas

Para estas funciones no va a importar el orden de declaración.

4.3 Declaración de funciones

4.3.1 Parámetros

El orden de declaración de los parámetros que no son por defecto va a ser en

orden decreciente de importancia. Por otro lado, en caso de que los parámetros sean por defecto, serán declarados primero los más propensos a ser utilizados.

Adicionalmente para los parámetros se debe:

- Proveer nombres formales en la declaración.

4.3.2 Aspectos de complejidad

La complejidad ciclomática tiene que ser menor que 10. Mientras que la complejidad ciclomática extendida que 15. El número máximo de líneas que puede contener el cuerpo de una función es de 60, y la cantidad máxima de puntos de retorno va a ser 5.

5 Base de Datos

5.1 Nombre para Base de Datos

La Base de Datos no requiere prefijo.

Las letras de la palabra estarán en todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, y la longitud de este no debe exceder los 20 caracteres.

5.2 Nombre para Tablas

Las Tablas requieren el prefijo **esp**.

Después del prefijo el resto de las letras estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, caracteres especiales, y la longitud de este no debe exceder los 20 caracteres.

5.3 Nombre para Campos

Los Campos no requieren prefijo.

Las letras de la palabra estarán todas en minúscula. Es necesario que el nombre indique propósito, que contenga letras, caracteres especiales, y la longitud de este no debe exceder los 20 caracteres.

Anexo 14. Características del sistema. COCOMO II.

Nombre	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
Insertar noticia	4	18	Alto
Autenticar usuario	1	3	Bajo

EI, entradas que le son proporcionadas al sistema.

Nombre	Cantidad de Ficheros	Cantidad de elementos de datos	Clasificación
Listar noticias internas	3	14	Medio
Mostrar noticias internas	1	2	Bajo

EO, salidas asociadas al sistema que tienen elementos de filtraje de información.

Nombre	Cantidad de ficheros	Cantidad de elementos de datos	Clasificación
esp_content	1	12	Bajo
esp_categories	1	8	Bajo
esp_sections	1	7	Bajo
esp_messages	1	3	Bajo
esp_users	1	5	Bajo
esp_core_acl_aro_groups	1	2	Bajo

EIF, ficheros lógicos o de almacenamiento de información externos al sistema.