

Universidad de las Ciencias Informáticas



Facultad 2

**TRABAJO DE DIPLOMA PARA OPTAR POR EL
TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS**

TÍTULO:

CLASIFICACIÓN AUTOMÁTICA DE LOS COMENTARIOS DE LOS
USUARIOS DE APLICACIONES DE SOFTWARE

AUTORAS:

LAZARA BARRIOS DOMINGUEZ
LISANDRA VERDEJA GARCÍA

TUTOR: ING. VLADIMIR MILIÁN NÚÑEZ

**“Año 58 de la Revolución”
La Habana, 2016**

PENSAMIENTO



“Después de escalar una montaña muy alta, descubrimos que hay muchas otras montañas por escalar.”

Nelson Mandela

DECLARACIÓN DE AUTORÍA

Declaramos ser autoras del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas con los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Lazara Barrios Dominguez

Lisandra Verdeja García

Firma del Autor

Firma del Autor

Vladimir Milián Núñez

Firma del Tutor

DATOS DE CONTACTO

Datos del Autor:

Lazara Barrios Dominguez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo: lbarrios@estudiantes.uci.cu

Datos del Autor:

Lisandra Verdeja García

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo: lverdeja@estudiantes.uci.cu

Datos del Tutor:

Ing. Vladimir Milián Núñez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo: vmilian@uci.cu

AGRADECIMIENTOS

Agradezco en primer lugar a mi tatica querido, la persona que más quiero en este mundo, que ha sido para mí, más que mi hermano, mi amigo, mi maestro, mi padre, mi confesor, mi enfermero, mi paño de lágrimas, mi guía, mi todo.

Le agradezco desde lo más profundo de mi corazón a mi mamita linda y a mi papito su entera dedicación, sus noches de desvelo mientras nos enfermábamos. Agradezco sus grandes esfuerzos para poder mantenernos a mi hermano y mí durante todos estos años de estudios.

A mimá porque ella al igual que mi mamá ha sido la persona que más me ha apoyado durante toda mi vida. Nunca olvidaré todas las cosas que ha hecho por mí, sus cuidados, sus atenciones, sus caricias, sus canciones y sus cuentos.

A pipo y a abuelo lele, que, aunque ahora no están presentes siempre velaron porque no nos faltara nada, ni a mí ni a mi familia, aun cuando ya no podían.

En general a toda mi familia por apoyarme siempre y no dejar que abandonara mis estudios, en especial a mi primo Erodito por ser el impulsor de mis sacrificios, gracias por estar en mi vida.

A papi Toni por estar siempre atento con todo como si yo fuera otra más de sus nietecitas.

A todos mis amigos de aula desde la primaria, secundaria, pre y especialmente a todas las personitas maravillosas que compartieron conmigo en esta Universidad, por siempre estar ahí cuando más los necesito, en especial a Midiala, a Bania, a Anita, a Yeyli, a Mamuchi, a Dania, a Juan Carlos, a Mare, a Leydis, a Aray, a Eloy, a Yandy (mi negrito), a Tuta, a Arle, a Dieguito, a Niharby, a Anyer, al Rafa, al Gera, a Daquito, a Alexander, a Migdalia, a Juan, a Pablo, a Maturell, a Wilfredo a los muchachos de la plena y a todo el equipo de kikimbo!, incluyendo a Pichardo y a Tan, muchas gracias por darme tantas alegrías.

A mi compañera de tesis porque sin ella no habría podido con todo, fue más que una amiga durante todo este tiempo, le agradezco ser tan especial y buena como es.

A Paneque, a Piillo, a José Javier, a Ernesto, a Rene, a Jéssica, a Gago y a todas esas personas que me explicaron incansables veces, muchas gracias por toda su paciencia y dedicación.

A mi novio Julio le quiero dar gracias por haber llegado a mi vida y por hacerme tan feliz.

Al tutor porque a pesar de que yo lo sacaba de sus casillas siempre nos apoyó.

A Mayda y a Edisnel gracias por dedicarnos parte de su tiempo, por sus consejos, su ayuda incondicional y su amabilidad.

Al oponente porque a pesar de no tener vínculo alguno con nosotras siempre fue muy atento y amable.

A todos mis profesores desde la primaria, en especial a Ángela, a Arlés, a Mamita, el Flori, al Dary (Abelito), a Omar Garrido, a Abel, a Franklin, Baby, a Yaniselis, a Deivys, a Iyatne y a Carlos Milán por todo el apoyo que me dieron, por su dedicación, su ayuda incondicional, su cariño y sus buenos consejos.

A todas las personas maravillosas que han formado parte de mi vida, que, aunque no las mencione aquí ocupan un lugar muy especial en mi corazón y jamás olvidaré todas las cosas buenas que hicieron por mí y los momentos inolvidables que pasamos juntos, a la Universidad y a la Facultad 2 por educarme como una verdadera profesional.

Lazara Barrios Dominguez

Agradezco a mis padres por darme su apoyo incondicional, amor, consejos, guiarme por el buen camino y hacer de mí una niña grande capaz de enfrentarse a sus propios retos con logros satisfactorios. Gracias a ellos por ser padres tan maravillosos y estar al pendiente en cada momento.

A mis hermanos Michel y Laura por ser tan geniales y brindarme ese cariño tan lindo que es el de hermanos.

A mi novio Jorge Eloy por estar durante estos cinco años de la carrera dándome amor, apoyo, comprensión y compañía conociendo los riesgos.

A mi bello sobrinito Carlos Michel por ser la alegría más grande de la familia.

A mis abuelas Olga, Zoila y Margot por sus ayudas, jocosidades y apoyo cuando lo he necesitado.

A mi tía Flor por aconsejarme, estar al pendiente y cuidarnos en momentos difíciles.

A mis tías y tíos Cachi, Cary, Marlen, Manolito y Gerardo por su apoyo y ayuda.

A mi madrina que a pesar de que no nos bautizó ha sido capaz de darnos mucho amor a mis hermanos y a mí.

A mis suegros por su ayuda, por adoptarme como su hija y darme un novio tan maravilloso.

A Gerardito que durante estos últimos tres años hemos sido un punto de apoyo el uno para el otro.

A Yandy por el tiempo, la dedicación y la valiosa ayuda que nos brindó con la tesis.

A mi compañera de tesis, y amiga Lazara por ser tan atenta conmigo, por hacerme reír cuando más enredadas estábamos con la tesis. Con ella conocí lo que es ser una persona humana, preocupada, sincera y capaz de esforzarse al máximo para dar lo mejor de sí.

A Marelys por ser tan cariñosa y enseñarme que tan sensible puede ser una persona.

A Aray por haber estado presente en mi vida durante estos cinco años en los que compartimos momentos buenos y malos.

A Ernesto por su compañía y el apoyo incondicional que me ha dado desde que llegó a la universidad.

A Alexander y a Manuel por el apoyo que me dieron en el primer año de la carrera, cuando se me hacía imposible adaptarme a la beca.

A Alexander, Gerardo, José Javier, Aray, Lazara por ser compañeros de estudio en trabajos y ante la llegada de las pruebas.

A René, Pablo, Ernesto, Jorge Eloy, Alexander y Sergio que aportaron su más humilde y desinteresada ayuda en la realización de la tesis, a pesar de que igual que nosotras tenían otras cosas por hacer.

Al tutor por habernos dado el tema de tesis cuando ya estábamos con todos los recursos agotados y guiarnos en el desarrollo de la tesis.

Al tribunal por habernos ayudado durante los seminarios de tesis especialmente a Mayda y Edisnel por todas las veces que necesitamos que nos guiara en el desarrollo de la misma.

A todos los profesores que me ayudaron en mi paso por la carrera especialmente a Siomara, Noel, Abel, Ariel, Iyatne, Yaniselis, Bárbara, Deivis.

A todas las amistades del aula y de la universidad que de una forma u otra hemos compartido momentos inolvidables.

En fin, agradezco a todas aquellas personas que conocí durante toda mi vida, que me extendieron la mano cuando más lo necesité.

Lisandra Verdeja García

DEDICATORIA

Dedico este trabajo de diploma desde lo más profundo de mi corazón a mis abuelos y a las personas que más quiero en este mundo, por las que tengo tantos sentimientos que son imposibles de describir, a mis padres y a mi hermano porque este triunfo no es mío, sino de ellos por llevarme por el buen camino, por estar presentes para mí y apoyarme siempre. Por llenar de alegrías todos los días de mi vida y confiar siempre en mí a pesar de todo.

Lazara Barrios Dominguez

Le dedico todo mi trabajo y mis esfuerzos durante mi vida como estudiante a mis padres, mis hermanos, mi bello sobrino y mi novio que juntos han hecho posible este sueño.

Lisandra Verdeja García

RESUMEN

Con el avance de las tecnologías de la información y las comunicaciones los comentarios de las personas representan un elemento esencial para la toma de decisiones, para que los especialistas y las empresas de desarrollo de software puedan mejorar sus productos. El objetivo de esta investigación es desarrollar una aplicación web que permita a los equipos de especialistas clasificar las opiniones de los usuarios en positivas, negativas o neutras. La herramienta informática desarrollada contribuye a optimizar el proceso de desarrollo de aplicaciones de software. Se proyectó sobre el marco de trabajo Django 1.6, mediante el lenguaje de programación Python 2.7 con el uso de las bibliotecas NLTK y TextBlob.

Palabras Clave: comentarios de usuarios, minería de opinión, minería de texto.

ÍNDICE DE CONTENIDO

PENSAMIENTO	1
DECLARACIÓN DE AUTORÍA	2
DATOS DE CONTACTO	3
AGRADECIMIENTOS	4
DEDICATORIA	6
RESUMEN	7
ÍNDICE DE CONTENIDO	8
ÍNDICE DE IMÁGENES	10
ÍNDICE DE TABLAS	11
INTRODUCCIÓN	12
Capítulo I. Fundamentación teórica	18
1.1 Introducción	18
1.2 Análisis de comentarios de usuarios y su impacto social	18
1.3 Conceptos asociados al objeto de estudio	21
1.4 Metodología de desarrollo	24
1.5 Lenguajes de programación	28
1.6 Marco de trabajo	30
1.7 Lenguaje de modelado.....	31
1.8 Herramientas de desarrollo	32
1.9 Conclusiones parciales	34
Capítulo 2. Características del sistema	35
2.1 Introducción	35
2.2 Propuesta de solución.....	35
2.3 Modelo de negocio	36
2.4 Levantamiento de requisitos.....	40
2.5 Modelo de casos de uso del sistema.....	43
2.6 Conclusiones parciales	44
Capítulo 3. Diseño del sistema	46
3.1 Introducción	46
3.2 Diagrama de paquetes	46
3.3 Diagrama de clases de diseño	47
3.4 Modelo físico de los datos	47
3.5 Diseño de la arquitectura	48
3.6 Patrón arquitectónico	49

3.7	Patrones de diseño	50
3.8	Conclusiones parciales	52
Capítulo 4. Implementación y prueba		53
4.1	Introducción	53
4.2	Diagrama de componente	53
4.3	Estándares de codificación.....	54
4.4	Diagrama de despliegue	55
4.5	Pruebas.....	56
4.6	Resultados de las pruebas.....	59
4.7	Conclusiones parciales	60
CONCLUSIONES GENERALES		61
RECOMENDACIONES		62
REFERENCIAS.....		63
BIBLIOGRAFÍA		67
ANEXOS.....		71

ÍNDICE DE IMÁGENES

Figura 1. Proceso KDD.....	21
Figura 2. Fases de CRIP-DM.	26
Figura 3. Fases y disciplinas de AUP.	28
Figura 4. Propuesta de solución	36
Figura 5. Diagrama de casos de uso del sistema	44
Figura 6. Diagrama de paquetes	46
Figura 7. Diagrama de clases del diseño, caso de uso: “Autenticar usuario”	47
Figura 8. Modelo físico de la base de datos	48
Figura 9. Arquitectura cliente-servidor	49
Figura 10. Patrón arquitectónico Modelo- Plantilla-Vista	50
Figura 11. Ejemplo de utilización del patrón Decorador	51
Figura 12. Ejemplo de utilización del patrón Alta Cohesión	51
Figura 13. Ejemplo de utilización del patrón Bajo Acoplamiento.....	51
Figura 14. Ejemplo de utilización del patrón Creador	52
Figura 15. Diagrama de componente	54
Figura 16. Diagrama de despliegue.....	56
Figura 17. Resultado de las pruebas realizadas	57
Figura 18. Reporte de la prueba de carga	60
Figura 19. Diagrama de clases del diseño “Registrar usuario”	82
Figura 20. Diagrama de clases del diseño “Gestionar documento”	82
Figura 21. Diagrama de clases del diseño “Mostrar resultados”	83

ÍNDICE DE TABLAS

Tabla 1. Descripción del “Proceso general”	37
Tabla 2. Descripción del proceso “Preparar datos”	38
Tabla 3. Descripción del proceso “Aplicar minería”	39
Tabla 4. Descripción del proceso “Evaluar comportamiento”	40
Tabla 5. Especificación del RF#1 “Autenticar usuario”	42
Tabla 6. Actor del sistema	44
Tabla 7. Caso de prueba del RF#1: “Autenticar usuario”	58
Tabla 8. Resultados de las pruebas	59
Tabla 9. Especificación del RF#2 “Registrar usuario”	72
Tabla 10. Especificación del RF#3 “Gestionar documento”	73
Tabla 11. Especificación del RF#3.1 “Crear documento”	74
Tabla 12. Especificación del RF#3.2 “Editar documento”	75
Tabla 13. Especificación del RF#3.3 “Eliminar documento”	75
Tabla 14. Especificación del RF#5 “Mostrar resultados”	77
Tabla 15. Especificación del RF#5.1 “Graficar categorías”	78
Tabla 16. Especificación del RF #5.2” Graficar comentarios”	79
Tabla 17. Especificación del RF #5.3 “Listar categorías”	80
Tabla 18. Especificación del RF #6 “Exportar resultados”	81
Tabla 19. Caso de prueba del RF#2: “Registrar usuario”	84
Tabla 20. Caso de prueba del RF#3.1: “Crear documento”	84
Tabla 21. Caso de prueba del RF#3.2: “Editar documento”	85

INTRODUCCIÓN

Los productos informáticos deben proporcionar una mejora en el desenvolvimiento, reducción de errores y ahorro de costos operativos para las empresas, debido a que el mercado de desarrollo de software es un entorno¹ muy competitivo. Es por ello que el desarrollo de una aplicación de software no termina con su liberación o disponibilidad a los clientes, pues las empresas y desarrolladores² necesitan mejorar sus aplicaciones con el fin de aumentar las ventas e ingresos.

Durante todo el proceso de desarrollo de la aplicación es necesario asegurar la calidad y robustez del software, ya sea a través de pruebas, encuestas³ o análisis de opiniones de usuarios. La realización de pruebas de software se utiliza para: detectar fallas o posibles errores en el sistema y para comprobar su rendimiento. A pesar de que ayudan al mejoramiento del sistema que se desarrolla, nunca alcanzan a exigir a una aplicación, como pueden hacerlo cientos de usuarios usándola en diferentes ambientes y en un grupo heterogéneo de dispositivos.

Además de las pruebas masivas de software, existe la retroalimentación de información sobre el uso de las aplicaciones informáticas por los usuarios mediante la realización de encuestas o pruebas de aceptación. Las encuestas son elaboradas por un grupo de expertos, asegurándose de que una muestra representativa de los usuarios ~~conteste~~[contesten](#) la misma con el objetivo de obtener conocimientos valiosos con el fin de maximizar el éxito. El esfuerzo detrás de este proceso no es despreciable y en muchas ocasiones no es factible, pues los usuarios no comprenden la necesidad de la veracidad de los datos⁴ con que llenan estas interrogantes.

Una tercera posibilidad es aprovechar una valiosa fuente de información⁵: las revisiones de los usuarios (comúnmente conocidas como *users reviews*). Existen varios portales en los cuales los usuarios de

¹ El entorno es aquello que rodea a algo o alguien.

² Alguien que programa ordenadores o diseña el sistema informático para que coincida con los requisitos de un analista de sistemas.

³ Es un procedimiento dentro de los diseños de una investigación descriptiva en el que el investigador busca recopilar datos por medio de un cuestionario previamente diseñado en dar una entrevista a alguien, sin modificar el entorno ni el fenómeno donde se recoge la información ya sea para entregarlo en forma de tríptico, gráfica o tabla.

⁴ Los datos describen hechos empíricos, sucesos y entidades.

⁵La información es un conjunto organizado de datos procesados.

una aplicación específica pueden dejar un comentario, mediante la asignación de una puntuación al sitio (generalmente expresado con un número de estrellas que van de uno a cinco). Además, cuentan con un campo de texto libre para informar errores, recomendar nuevas características, o simplemente describir sus sentimientos y experiencias durante el uso de la aplicación.

Estas opiniones, escritas en un lenguaje coloquial y natural, representan una fuente valiosa de información para los creadores de software. Su análisis permite obtener recomendaciones sobre cómo mejorar la aplicación y entender lo que necesitan los usuarios (Allic, 2010). Esta información puede ser muy útil para los desarrolladores cuando se desea liberar una nueva versión de su aplicación o para estudiar la competencia.

Algunas aplicaciones, como Google Play⁶ proporcionan apoyo a los desarrolladores con la posibilidad de clasificar las revisiones por calificación para detectar fácilmente los más críticos. Sin embargo, algunos usuarios simplemente asignan puntuaciones muy bajas (por ejemplo, una estrella) sólo para aumentar la visibilidad de su revisión y en muchos casos es más positivo que negativo su contenido. Esto reduce claramente la visibilidad de otros comentarios que podrían contener información sobre errores, ocultándolos de los ojos de los desarrolladores. Por lo tanto, este apoyo trivial no es suficiente.

Con el fin de obtener ventajas de manera efectiva de la información que los usuarios dejan en las opiniones, un desarrollador tiene que navegar a través de todo el conjunto de comentarios, leer manualmente cada revisión y comprobar si contiene información útil (por ejemplo, sugerencias para nuevas características o de mejoras). Este proceso no es factible si la aplicación recibe cientos de comentarios por día, como sucede con las aplicaciones populares.

Por otra parte, alrededor del 70% de las opiniones de los usuarios no resultan útiles para los desarrolladores- porque muchas se repiten, son similares o simplemente son comentarios neutros sobre el uso de la aplicación (Jojo, 2011). Dicha situación conlleva a que los desarrolladores deben distinguir entre la información útil y no útil, aumentando el tiempo en el análisis de esta información y alargando el tiempo de liberación de la(s) versión(es) planificada(s). Además, al no tener presente las opiniones de los usuarios pueden liberarse nuevas versiones de la aplicación con características no deseadas o con errores, lo que resultaría una posible pérdida de nuevos o antiguos usuarios, e incluso la correspondiente pérdida monetaria.

⁶ <https://play.google.com/store?hl=es>

Partiendo de la situación problemática expuesta anteriormente surge el siguiente **problema a resolver**:
¿Cómo contribuir a la obtención de información útil a partir de la clasificación automática de comentarios de los usuarios de aplicaciones de software?

Teniendo como **objeto de investigación**: la minería de texto.

El **campo de acción** se centrará en: la minería de opinión atendiendo a los comentarios de usuarios.

Para dar solución al problema a resolver se define como **objetivo general**: Desarrollar una herramienta⁷ de clasificación automática de información útil a partir de comentarios de usuarios para apoyar el proceso de desarrollo de aplicaciones de software.

Para dar cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

1. Caracterizar los elementos teóricos-conceptuales referentes a la minería de texto y la minería de opinión.
2. Implementar una herramienta de clasificación automática de información.
3. Evaluar el modelo⁸ para el análisis de opiniones de usuarios.

Las **tareas de investigación** que se deben cumplir son:

1. Diagnóstico de la implementación de la minería de texto y la minería de opinión.
2. Análisis de tendencia y estado actual de la minería de opinión aplicado a los comentarios de los usuarios.
3. Selección de la metodología y las herramientas para el desarrollo del modelo.
4. Definición de los requisitos funcionales y no funcionales.
5. Descripción de los requisitos funcionales y no funcionales.
6. Diseño de una herramienta para la clasificación automática de los comentarios de los usuarios.

⁷ Se usa para facilitar la realización de una actividad cualquiera.

⁸ Descripción de cómo se representan los datos, sea en una empresa, en un sistema de información o en un sistema de gestión de base de datos.

7. Implementación de una herramienta para la clasificación automática de los comentarios de los usuarios.
8. Evaluación del modelo obtenido con varios conjuntos de datos de prueba.
9. Realización de pruebas de software al sistema desarrollado.

Se espera obtener una aplicación que sea capaz de clasificar las opiniones de los usuarios para que los desarrolladores de software tengan en cuenta los *users reviews*. Los comentarios de usuarios constituyen una ayuda para que los integrantes de los equipos de desarrollo conozcan las opiniones de los clientes, relacionados con las funcionalidades del software, permitiéndoles mejorar el producto y erradicar no conformidades.

Inicialmente se utilizará un dataset (conjunto de datos) sobre Norton⁹ extraído de internet para probar la aplicación web que será creada, producto de la falta de información que existe en la Universidad de las Ciencias Informáticas (UCI) sobre los productos que en ella se crean, pues sólo se generan reportes¹⁰ de errores y no se tienen en cuenta los comentarios de usuarios que explotan sus funcionalidades.

Para dar cumplimiento a las tareas investigativas se emplearon los siguientes **métodos científicos de investigación**:

Métodos teóricos:

- **Histórico-Lógico:** se empleará para estudiar la trayectoria y el desarrollo histórico de las herramientas que utilizan el análisis de comentarios de usuarios y poder comprender la lógica de sus aportes, así como las tendencias actuales.
- **Analítico-Sintético:** se utilizará para analizar la documentación recopilada sobre la minería de textos y sus técnicas, seleccionando los elementos esenciales para darle solución al problema existente.
- **Modelación:** se utilizará para modelar el proceso de desarrollo de la investigación mediante

⁹ <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>

¹⁰ Un reporte es un informe o una noticia. Este tipo de documento (que puede ser impreso, digital, audiovisual, etc.) pretende transmitir una información.

los diagramas¹¹ correspondientes a la fase de análisis y diseño, proporcionando una mejor perspectiva de la aplicación a desarrollar.

Métodos empíricos:

- **Entrevista**¹²: se utilizará la técnica de entrevista personal a especialistas¹³ del Centro de Soporte Técnico de la Universidad de las Ciencias Informáticas con el objetivo de conocer cómo gestionan, monitorean y controlan los servicios que brindan. Ver Anexo #1. Entrevista realizada al Director del Centro de Soporte de la UCI.

El presente trabajo de diploma se divide en:

Capítulo 1. “Fundamentación teórica”

En este capítulo se describen los conceptos asociados al objeto de estudio: el proceso de descubrimiento de conocimiento (KDD), la minería de texto, la minería de opinión y el procesamiento de lenguaje natural. Se detallan además las técnicas de la minería de texto. Se realizará un estudio de las herramientas que emplean el análisis de comentarios de usuarios. Se definirá la metodología a emplear, el (lenguaje de programación)¹⁴ a utilizar y las herramientas necesarias para realizar el presente trabajo de diploma.

Capítulo 2. “Características del sistema”

En este capítulo se expone la propuesta de solución del sistema a desarrollar. Se abordan los procesos¹⁵ de negocio¹⁶, teniendo en cuenta los diagramas y sus descripciones. Se definen los

¹¹ Un diagrama es un tipo de esquema de información que representa datos numéricos tabulados.

¹² Reunión de dos o más personas para tratar algún asunto, generalmente profesional o de negocios.

¹³ Un especialista es una persona que cultiva o practica una rama determinada de un arte o una ciencia.

¹⁴ Es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

¹⁵ Un proceso es un conjunto de actividades mutuamente relacionadas o que, al interactuar juntas, simultánea o sucesivamente en los elementos de entrada los convierten en productos o resultados.

¹⁶ Actividad comercial o social que se ha pensado y que se desea desarrollar.

requisitos funcionales y no funcionales con los que debe contar la aplicación, además de ilustrarse el caso de uso del sistema.

Capítulo 3. “Diseño del sistema”

En este capítulo se propone el diseño del sistema AOpinion, a partir del modelo físico de datos, el diagrama de paquetes y de los diagramas de clases del diseño con sus relaciones. Se muestra la arquitectura, el patrón arquitectónico y los patrones¹⁷ de diseño utilizados en la aplicación.

Capítulo 4. “Implementación y prueba”

En este capítulo se mostrará la situación física de los componentes lógicos desarrollados con el diagrama de despliegue, los elementos físicos que integran el sistema y las relaciones que existen entre ellos. Se describen los principales resultados obtenidos en la etapa de prueba.

¹⁷ Conjunto de reglas que pueden ser usadas para crear o generar entidades.

Capítulo I. Fundamentación teórica

1.1 Introducción

En este capítulo se plantean los elementos básicos del estudio del tema en cuestión a través de la indagación bibliográfica. A continuación, se abordan los principales conceptos que serán tratados durante el desarrollo del trabajo. Se realiza un análisis de las diferentes tendencias que delinean el estado actual de la minería de opinión aplicado a los comentarios de usuarios. Además, se muestra una caracterización de: las técnicas, las tecnologías, los lenguajes y metodologías seleccionadas para el desarrollo del sistema.

1.2 Análisis de comentarios de usuarios y su impacto social

Las opiniones son expresiones subjetivas que describen los sentimientos de las personas hacia otros individuos, entidades, eventos, objetos y sus propiedades. El análisis de sentimientos es un área de investigación que se enfoca en las opiniones. Se trata primordialmente de un problema de clasificación en el que se intenta saber si se emite una opinión positiva, negativa o neutra (Liu, 2010).

Hace algunos años no se podía conocer el estado de ánimo de una persona de forma automática sin necesitar de la participación directa del sujeto. Gracias a los avances de la ciencia y la tecnología existen diversos sitios en internet que permiten dejar plasmado lo que piensan las personas sobre determinados campos. Ejemplo de ello son: las compras por internet para adquirir y ver cuál es el mejor producto, en entretenimiento para ver las películas y los programas más populares, en el gobierno para ayudar a los políticos a conocer sus fortalezas y debilidades en campañas electorales, en el desarrollo de software para conocer los elementos positivos, negativos y neutros (Binali, y otros, 2009).

Los comentarios de usuarios son una herramienta de gran ayuda para estudiar lo que desean y piensan los usuarios en los lanzamientos de productos de software y así contribuir a desarrollar aplicaciones de mejor calidad con menor tiempo de respuesta. Brindarles a los usuarios la posibilidad de emitir sus opiniones, permite procesar y extraer información que es realmente importante, descubriendo conocimiento y patrones en estos datos (Minería de Datos como soporte a la toma de decisiones empresarial, 2007).

Estos autores abordan con gran precisión el significado de considerar los comentarios de usuarios como expresiones escritas en lenguaje natural, que expresan experiencias adquiridas de los clientes en la interacción con un software determinado. Sus ideas constituyen herramientas de gran ayuda cuando se desea conocer el criterio de los clientes en los lanzamientos de software. A continuación, se

describe la investigación en el ámbito internacional y nacional, respecto a sistemas relacionados con el campo de acción predeterminado, obteniendo los resultados que se muestran en el siguiente epígrafe.

1.2.1 Investigación de las Herramientas de Análisis de Comentarios

Google PLAY (tienda oficial de Google para terminales Android, distribuye películas, música, libros y, sobre todo, aplicaciones) sirve de apoyo a los desarrolladores, pues posibilita la clasificación de las revisiones por calificación para detectar fácilmente los más críticos. Sin embargo, algunos usuarios asignan puntuaciones muy bajas sólo para aumentar la visibilidad de su revisión y en muchos casos es más positiva que negativa su contenido (Crider, 2014). Esto reduce claramente la visibilidad de otras críticas que podrían contener información valiosa acerca de los errores, mal funcionamiento o mejoras, ocultándolos de los ojos de los desarrolladores. Por lo tanto, este apoyo no es suficiente. Otras aplicaciones parecidas son App Store¹⁸(servicio para que los usuarios busquen y descarguen aplicaciones informáticas) y AndroidPit¹⁹(permite realizar búsquedas de aplicaciones, ofrece numerosas recomendaciones y reviews(revisiones) propias).

Brandwatch es una herramienta de monitoreo²⁰ de redes sociales extremadamente útil pues permite obtener una idea de la opinión pública general sobre ciertos temas. Utiliza un proceso de reglas para ayudar a entender mejor las diferentes maneras en que el contexto puede afectar al sentimiento. Es una herramienta abarcadora, el precio más básico es de 800 dólares y en la medición de ‘sentimiento’ no es posible conocer en más detalle al usuario que emitió el comentario negativo o positivo (Mendoza, Milagros, 2014).

Otras aplicaciones semejantes son HootSuite(herramienta gratuita disponible para la gestión de las redes sociales, ya que cubre múltiples plataformas tales como: Twitter, Facebook, LinkedIn, WordPress, Foursquare y Google+), Social Mention (una de las mejores herramientas gratuitas de monitorización disponible en el mercado, pues analiza la información de una manera más profunda y también mide la influencia en cuatro categorías: Strenght (Fortaleza), Sentiment (Sentimiento), Passion (Pasión) y Reach (Alcance (2013).

¹⁸ <http://www.app-store.es/>

¹⁹ <http://www.androidpit.es/aplicaciones-android>

²⁰ Monitoreo es el proceso sistemático de recolectar, analizar y utilizar información para hacer seguimiento al progreso de un programa en pos de la consecución de sus objetivos, y para guiar las decisiones de gestión.

ITelligent, es un sistema de minería de opinión para inteligencia comercial; **OPAL**, plugin de Drupal que analiza comentarios realizados por los usuarios y detecta si el resultado es positivo, negativo o neutral; **IIC-Lynguoque**, es un conjunto de herramientas que ayudan a extraer la opinión positiva, negativa y neutra de un texto; **netOpinion**, permite conocer opiniones de productos o servicios en foros y redes sociales; **WebOpinion**, gestiona mes a mes la evolución de la imagen de un usuario en la red; y **Sentitex**, que consiste en un conjunto de aplicaciones para el análisis de sentimientos en textos. Desafortunadamente la mayoría de estas herramientas son propietarias y es necesario pagar la licencia para su utilización, ya que fueron concebidas con fines comerciales. Adicionalmente, cada una de ellas responde a objetivos específicos de la minería de opinión (PosNeg opinion: Una herramienta para gestionar comentarios de la web., 2015).

PosNeg Opinion es una herramienta desarrollada por el Departamento de Ciencia de la Computación en la Universidad Central de las Villas, para que el usuario analice un gran cúmulo de opiniones de manera sencilla, ya que se convierten los ficheros XML²¹ a texto plano y se analizan independientemente las opiniones almacenadas en una lista. Fue desarrollada completamente en JAVA, por lo que es multiplataforma, necesita como entrada un fichero XML con todas las opiniones a analizar y como salida muestra cuántas fueron positivas y cuántas negativas. A petición del usuario también retorna el porcentaje, además de una lista con las opiniones negativas y positivas (PosNeg opinion: Una herramienta para gestionar comentarios de la web., 2015).

Una vez descritas las herramientas antes expuestas se puede concluir que varias son propietarias y resulta necesario pagar la licencia o altos precios para su utilización, debido a que fueron concebidas con fines comerciales. Las herramientas de monitoreo de redes sociales como Brandwatch son demasiadas abarcadoras y algunas de ellas no controlan los artículos de noticias generales o foros, pues se quiere una herramienta centrada en el análisis de comentarios de usuarios sobre aplicaciones de software en el idioma español. PosNeg Opinion es una aplicación de escritorio para devolver las opiniones negativas y positivas, sin tener en cuenta clasificaciones específicas.

Por estas razones el objetivo del trabajo es desarrollar una aplicación que permita detectar de manera supervisada la polaridad de opiniones, reutilizando las bibliotecas de Python, es decir realizar una aplicación web donde se clasifiquen, grafiquen y filtren los comentarios de los usuarios por categorías subidos de un archivo .txt.

²¹ Lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

1.3 Conceptos asociados al objeto de estudio

1.3.1 Proceso de descubrimiento de conocimiento

El proceso de descubrimiento de conocimiento (KDD, por sus siglas del inglés *Knowledge Discovery in Databases*) se define como el proceso no trivial de identificar patrones con cierto grado de incertidumbre, que aporten información desconocida y útil tanto para el sistema, como para el usuario, y en última instancia comprensibles a partir de los datos...se trata de un proceso complejo que tiene como objetivo descubrir conocimiento útil (Hernández Orallo, y otros, 2004). El proceso KDD está organizado entorno a cinco fases (Figura 1):

1. **Integración y recopilación de datos:** se determinan las fuentes de información que pueden ser útiles, seguidamente se transforman los datos a un formato común, unificando toda la información recogida, detectando y resolviendo inconsistencias.
2. **Selección, limpieza y transformación:** en esta fase se eliminan o corrigen los datos incorrectos y se decide la estrategia a seguir con los datos incompletos.
3. **Minería de datos:** se decide cuál es la tarea a realizar y se elige el método que se va a utilizar.
4. **Evaluación e interpretación:** se evalúan los patrones y se analizan los expertos.
5. **Difusión y uso:** en esta última fase se hace uso del nuevo conocimiento y se hace participe de él a todos los posibles usuarios.

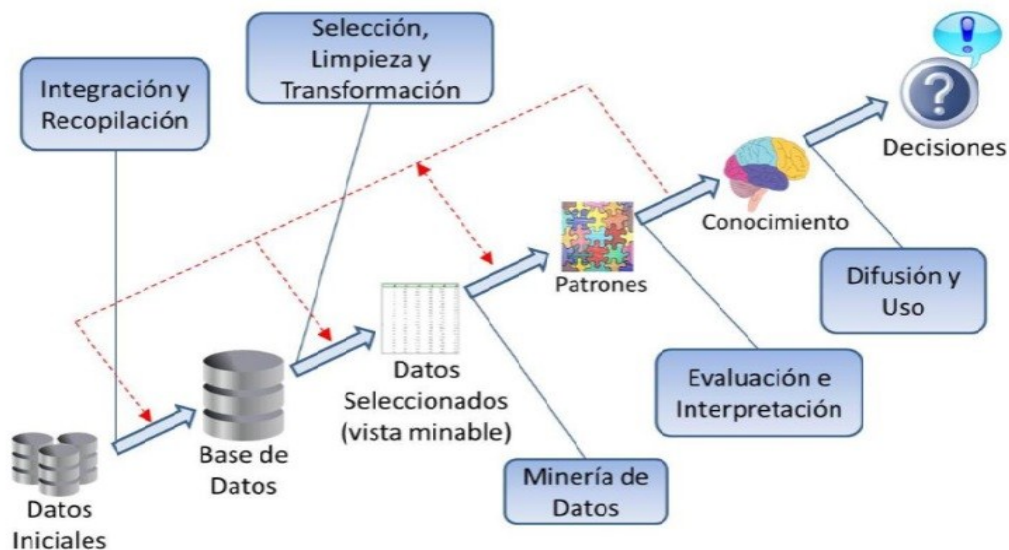


Figura 1. Proceso KDD

1.3.2 Minería de texto

La minería de texto un proceso que consiste en extraer información útil de conjuntos de documentos no estructurados de texto, y en identificar automáticamente patrones interesantes no triviales o conocimiento (Feldman, y otros, 1995) (Feldman, y otros, 2007).

Existen cuatro áreas de la minería de texto, según Feldman y Sanger la dividen en:

- Las tareas de pre-procesamiento incluyen todas aquellas rutinas, procesos y métodos requeridos para obtener y preparar los datos para las operaciones de descubrimiento de conocimiento.
- Las operaciones de minería incluyen la detección de patrones, análisis de tendencias y algoritmos de descubrimiento de conocimiento.
- La presentación y navegación incluye herramientas de visualización y navegación de los patrones orientados al usuario.
- Las técnicas de refinamiento incluyen los métodos que filtran información redundante, la ordenan, resumen, generalizan y agrupan.

1.3.2.1 Técnicas de la Minería de texto

Las técnicas de minería de texto se pueden clasificar en descriptivas y predictivas. Las descriptivas caracterizan las propiedades generales de los datos que se encuentran en las formas intermedias, mientras que las predictivas realizan inferencias en los datos para poder realizar predicciones. Hay que tener en cuenta lo que se desea obtener, para determinar cuál enfoque será usado.

Técnicas predictivas o supervisa.

Clasificación: Es el proceso de dividir un conjunto de datos en grupos mutuamente excluyentes de tal manera que cada miembro de un grupo esté lo "más cercano" posible a otro, y grupos diferentes estén lo "más lejos" posible uno del otro, donde la distancia está medida con respecto a variable(s) específica(s) las cuales se están tratando de predecir. Para este tipo de técnica existen algoritmos como Naive Bayes fundamentado en el teorema de Bayes (Hernández Orallo, y otros, 2004) (Vallejos, 2006).

Algoritmo de clasificación

La mayoría de los algoritmos de aprendizaje supervisado existentes pueden ser aplicados a la clasificación de polaridad de opiniones. El más utilizado debido a su simplicidad es el clasificador *Naive Bayes*, que consiste en calcular, para una instancia dada sin clasificar, cuál es la probabilidad de que se le asigne cada una de las clases y seleccionar la de mayor probabilidad. Se basa en la regla de *Bayes* y que "ingenuamente" asume la independencia de los atributos dada la clase.

Este método funciona bien con bases de datos reales, sobre todo cuando se combina con otros procedimientos de selección de atributos que sirven para eliminar la redundancia...el fundamento principal del clasificador *Naive Bayes* es la suposición de que los atributos son independientes conocido el valor de la variable clase. A pesar de que asumir esta suposición en el clasificador es sin duda bastante fuerte y poco realista en la mayoría de los casos se trata de unos de los clasificadores más utilizados (Hernández Orallo, y otros, 2004).

1.3.3 Minería de opinión

La minería de opinión es un área de la minería de textos que consiste en la clasificación de palabras, textos o documentos de acuerdo a las opiniones, sentimientos, emociones y subjetividades expresadas (Sentiment Analysis and Opinion Mining: A Survey, 2012).

La minería de opinión, también conocida como análisis de sentimientos es una rama del procesamiento de lenguaje natural de muy reciente aparición que se enfoca en las opiniones. Se trata primordialmente de un problema de clasificación de texto en el que se intenta saber si este refleja una opinión positiva, negativa o neutra (Liu, 2010) (Opinion Mining and SentimentAnalysis, 2008).

La minería de opinión es un campo relacionado con el análisis de los componentes subjetivos que están implícitos en los contenidos generados por los usuarios. Dentro de este campo, existen aplicaciones que realizan un análisis más o menos profundo de los contenidos textuales, en función del problema que se quiera resolver. Existen dos tipos de tareas relacionadas con la minería de opinión (Cortizo Pérez, 2015):

1. Detección de la polaridad: equivale a ser capaces de definir si una opinión es positiva o negativa. Más allá de una polaridad básica, también se puede querer obtener un valor numérico dentro de un rango específico, que de cierta forma permita acceder a la evaluación de una opinión determinada.
2. Análisis del sentimiento basado en características: se basa en determinar las distintas características del producto tratadas en la opinión escrita por el usuario, y para cada una de esas características mencionadas en la opinión, ser capaces de extraer de ellas una polaridad.

Este tipo de acercamiento es mucho más complejo y de un grado mucho más fino que la detección de la polaridad.

1.3.4 Procesamiento de lenguaje natural (NLP, por sus siglas del inglés *Natural Language Processing*)

El lenguaje natural se entiende como el lenguaje hablado y escrito con el propósito que exista comunicación entre una o varias personas, es más directo para expresar lo que se quiere comunicar (Benavides Cañón, y otros).

El procesamiento del lenguaje natural es una disciplina que relaciona directamente la computación y la lingüística, tiene como principal objetivo conseguir que el lenguaje humano pueda utilizarse como entrada en un proceso automatizado (Hernández, 2014).

Es un área de investigación en continuo desarrollo, se aplica en la actualidad en diferentes actividades como son la traducción automática, sistemas de recuperación de información, elaboración automática de resúmenes, interfaces en lenguaje natural (Aplicaciones del procesamiento del lenguaje natural en la recuperación de información en español., 2005).

1.4 Metodología de desarrollo

Una metodología consiste en un conjunto de actividades organizadas que tienen como objetivo la realización de un trabajo. Para cada actividad se define, además de las entradas y salidas, la forma en la que debe llevarse a cabo (Moine, 2013).

1.4.1 Metodología para enfrentar un proceso KDD

En un proyecto de KDD la mayor parte del esfuerzo se produce en la comprensión del negocio y la preparación de los datos, lo que unido al hecho de que no se puede arribar a conclusiones por adelantado, normalmente provoca retrasos y desviaciones en la planificación inicial. De aquí surge la necesidad, en muchos casos, de utilizar alguna metodología que guíe la ejecución del proyecto y facilite la planificación, dirección y seguimiento del mismo.

Se decidió utilizar la metodología CRISP-DM²² para modelar el proceso de minería de datos, debido a que fue diseñada de forma genérica, sin importar las herramientas que se utilicen para el desarrollo del proyecto. Su distribución es libre, encontrándose en constante desarrollo por la comunidad

²²Del inglés: Cross Industry Standard Process for Data Mining.

internacional, concibe el proyecto de KDD de forma global y estrechamente relacionado [al con el](#) negocio en cuestión. Además, propone un preciso y sólido repertorio de tareas de propósitos generales, por lo que goza de una importante popularidad, siendo, por tanto, frecuentemente empleada.

CRISP-DM

Método desarrollado con el objetivo de orientar trabajos de minería de datos. Como metodología incluye descripciones y las principales tareas de las fases de un proyecto, así como una explicación de la relación que existe entre las tareas.

CRISP-DM es flexible y se puede personalizar fácilmente, es decir, permite crear un modelo de minería de datos que se adapte a necesidades concretas. Además, profundiza en mayor detalle sobre las tareas y actividades a ejecutar en cada etapa del proceso de minería de datos (IBM, 2012).

CRISP-DM está estructurado en seis fases, algunas de estas son bidireccionales, es decir permitirá revisar parcial o totalmente las fases anteriores.

1. **Comprensión del negocio:** Se determinan los objetivos y requerimientos del negocio, se evalúa la situación para definir el problema como un proyecto de Minería de Datos y se elabora el plan del proyecto.
2. **Comprensión de los Datos:** Se recopilan los datos iniciales, se describen, se verifica la calidad de los mismos y se exploran para detectar los primeros puntos de vista o subconjuntos de carácter interesante que permitan crear hipótesis sobre la información oculta.
3. **Preparación de los Datos:** Se seleccionan los datos, se construyen, integran y estructuran para ser minados. Estas tareas incluyen tablas, registros, selección de atributos, así como la transformación y limpieza de los datos para que sean utilizados en las herramientas de modelado.
4. **Modelado:** Se seleccionan y aplican las técnicas de modelado, se genera el diseño del experimento y se construyen y evalúan los modelos.
5. **Evaluación:** Se evalúa a fondo el modelo revisando el proceso para construirlo, se asegura que el modelo cumple apropiadamente con los objetivos del negocio, se determinan temas relevantes en el negocio que no hayan sido suficientemente considerados y se valoran los resultados.

6. **Despliegue:** Se traza la estrategia ²³de empleo de los resultados, se planifica el mantenimiento del proceso, se organiza y presenta el conocimiento obtenido como resultado de manera que el cliente pueda usarlo, y se documentan las experiencias.

En la Figura 2, se puede apreciar las fases de CRISP-DM; en numerosas ocasiones es necesario volver atrás para reanalizar los resultados obtenidos. A la vez, el proceso se torna cíclico, debido a que no se termina con el despliegue de la solución, al contrario, el conocimiento obtenido puede provocar nuevas preguntas enfocadas en el negocio, donde los procesos de minería subsecuentes se beneficiarán de las experiencias anteriores (IBM, 2012).

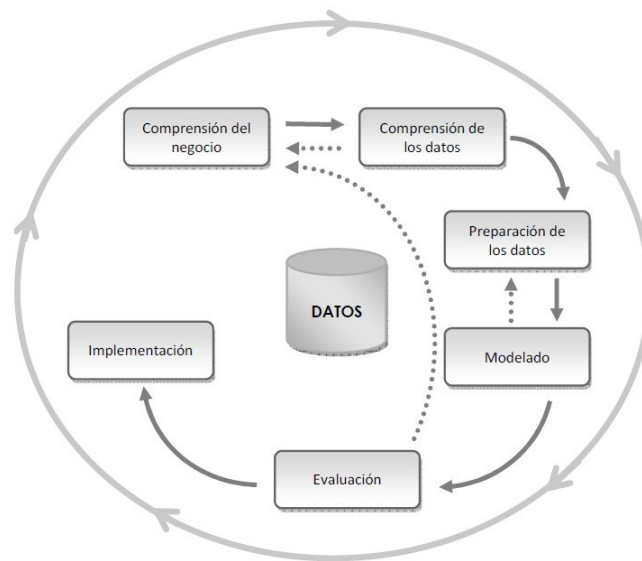


Figura 2. Fases de CRIP-DM.

1.4.2 Metodología de desarrollo del software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos informáticos. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a actividades del proceso de desarrollo (JACOBSON, y otros, 2000).

Se decide emplear para el desarrollo de la aplicación web la metodología AUP ya que es una metodología flexible que no requiere de una gran cantidad de desarrolladores. Es concisa en el aspecto de la documentación, permitiendo generar solo la necesaria. Está diseñada para trabajar en proyectos

²³ Conjunto de acciones que alinean las metas y objetivos de una organización

pequeños donde la atención se centra en las actividades que realmente son importantes. Permite el uso de herramientas de cualquier tipo, incluyendo las de código²⁴ abierto. Es fácil de manejar a través de herramientas de edición HTML sin necesidad de ser adaptada y es una metodología que se ajusta y aprovecha las ventajas que brindan las metodologías ágiles (Metodologías Ágiles. Proceso Unificado Ágil (AUP)).

AUP

Agile Unified Process (AUP) en español Proceso Unificado Ágil, es una versión simplificada del Proceso Racional Unificado (RUP). Descrito en una forma simple, fácil de entender y brinda un enfoque de desarrollo de software utilizando técnicas ágiles y conceptos del RUP. El proceso unificado (*Unified Process* o UP) es un marco de desarrollo de software iterativo e incremental (Metodologías Ágiles. Proceso Unificado Ágil (AUP)).

AUP abarca siete flujos de trabajos, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente. Dispone de cuatro fases igual que RUP: Creación, Elaboración, Construcción y Transición.

El ciclo de vida de AUP se puede ver en sus cuatro fases y su naturaleza iterativa se expresa en sus siete disciplinas:

1. **Modelado:** Comprender el negocio de la organización y el dominio del problema abordado por el proyecto, e identificar una solución al mismo que sea viable.
2. **Implementación:** Transformar el modelo realizado en código ejecutable y realizar pruebas de nivel básico, en particular pruebas unitarias.
3. **Prueba:** Realizar una evaluación objetiva para asegurar la calidad. Esto incluye buscar defectos, validar que el sistema funcione como debería, y verificar que se cumplan los requerimientos.
4. **Despliegue:** Planificar la liberación del sistema.
5. **Gestión de configuración:** Administrar el acceso a los artefactos del proyecto.
6. **Gestión de proyectos:** Dirigir las actividades que forman parte del proyecto.
7. **Ambiente:** Facilitar todo el entorno de manera tal que permita el desarrollo del proyecto.

AUP divide el ciclo de desarrollo en 4 fases:

²⁴ Es el texto escrito en un lenguaje de programación que ha de ser compilado o interpretado para ejecutarse en una computadora.

- **Inicio:** Identificación del alcance y dimensión del proyecto, propuesta de la arquitectura y el presupuesto del cliente.
- **Elaboración:** Confirmación de la capacidad de la arquitectura.
- **Construcción:** Desarrollo incremental del sistema, siguiendo las prioridades funcionales de los implicados.
- **Transición:** Validación e implantación del sistema.

AUP es una metodología de desarrollo ágil, la cual se basa en los siguientes principios:

- El personal sabe lo que está haciendo: la gente no va a leer detalladamente el proceso de documentación, pero algunos quieren una orientación de alto nivel o formación de vez en cuando.
- Simplicidad: todo se describe brevemente utilizando un conjunto de páginas, no miles de ellas.
- Agilidad: el ajuste a los valores y principios de la Alianza Ágil.
- Centrarse en actividades de alto valor: la atención se centra en las actividades esenciales para el desarrollo, no todas las actividades que suceden forman parte del proyecto.

En la figura 3, se muestra la interrelación entre las fases y las disciplinas propuestas por la metodología.

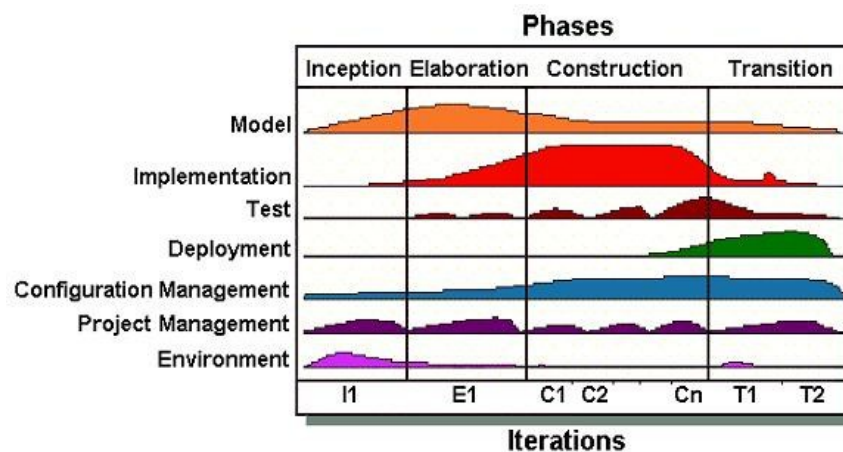


Figura 3. Fases y disciplinas de AUP.

1.5 Lenguajes de programación

1.5.1 Python 2.7

Lenguaje interpretado o de script que tiene una sintaxis simple, clara y sencilla. Python usa tipado dinámico, pues no es necesario declarar el tipo de datos que va a contener una determinada variable. Es multiplataforma y tiene gran cantidad de librerías disponibles (González Duque, 2015). Se utilizarán las siguientes bibliotecas:

- NLTK²⁵ (*Natural Language Toolkit*): conjunto de técnicas que permiten el análisis y manipulación del lenguaje natural. Se utiliza para la creación de programas en Python que interpretan el lenguaje humano. Permite realizar tareas de transformación y limpieza de documentos tales como: eliminar caracteres especiales, signos de puntuación, convertir todo el texto en minúscula, eliminar palabras comunes o sin significado (conocidas como *stopwords*) de la lengua en la que está escrito tales como: el, para, de, por, y, un, entre otras.
- TextBlob:²⁶ es una biblioteca de Python para el procesamiento de datos textuales, implementada sobre NLTK proporcionando una API simple para el procesamiento de lenguaje natural (NLP). Entre sus características se destacan la extracción de sintagma nominal, el análisis de los sentimientos, la clasificación, la traducción de idiomas y la detección impulsado por Google Translate²⁷ (Traductor de google). Además, Tokenization (División del texto en palabras y oraciones), la palabra de inflexión (pluralización y la singularización), lematización (llevar la palabra a la raíz) y corrección ortográfica.
- Psycopg:²⁸ establece la conexión entre python y el gestor de base de datos PostgreSQL.

1.5.2 HTML 5.0

HTML 5 es una nueva versión con elementos, atributos, comportamientos y un conjunto amplio de tecnologías que permiten crear sitios web y diversas aplicaciones de gran alcance. Es un lenguaje de marca usado para estructurar y presentar el contenido para la web (Álvarez, Miguel Ángel, 2009).

1.5.3 CSS3

Las hojas de estilo en cascada (Cascading Style Sheets, CSS por sus siglas en inglés) son las que ofrecen la posibilidad de definir las reglas y estilos de representación en diferentes dispositivos, ya sean pantallas de equipos de escritorio, portátiles, móviles, impresoras u otros dispositivos capaces de mostrar contenidos web. Permiten definir de manera eficiente la representación de las páginas, además es uno de los conocimientos fundamentales que todo diseñador web debe manejar a la perfección para realizar su trabajo (Damián, 2010).

²⁵ <http://www.nltk.org/>

²⁶ <https://pypi.python.org/pypi/textblob>

²⁷ <http://translate.google.com.cu>

²⁸ <https://pypi.python.org/pypi/psycopg2>

1.5.4 JavaScript

Javascript es un lenguaje que puede ser utilizado por profesionales y por quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, pues los [navegadores](#) son los encargados de interpretar estos códigos. Es muy utilizado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador web, así como para validar datos de entrada en las interfaces de las aplicaciones (Pérez Valdés, Damián, 2007). Como librerías utilizadas en JavaScript se encuentran:

- **Jspdf:** es una librería para Java Script que permite generar documentos PDF a partir de una plantilla HTML o directamente por programación. Esto permite poder diseñar un documento PDF de manera sencilla y adaptable. Una ventaja importante que se obtiene al utilizar esta librería es la de no utilizar recursos en el servidor ya que sólo se ejecuta del lado del cliente (navegador) **(Di Meglio, José , 2015)**.
- **HighCharts:** es una librería escrita en Javascript que permite la creación de gráficas. Ofrece un método fácil e interactivo para insertar [graficas](#) en el sitio web. Es compatible con todos los navegadores modernos, no es comercial, no se necesita el permiso de los autores para su implementación en sitios web personales o sin fines de lucro. Es abierto, todas las características pueden ser personalizadas permitiendo una gran flexibilidad, además está escrito solamente con código Javascript, sólo se requiere incluir el archivo highcharts.js y cualquiera de los tres frameworks más populares de Javascript (jQuery, MooTools o Prototype) (2013).
- **jQuery:**^{29 30} librería de JavaScript en su versión 1.8.2 que permite programar páginas dinámicas compatibles con todos los navegadores. Facilita el desarrollo de aplicaciones enriquecidas del lado del cliente, en Javascript, compatible con todos los navegadores.

1.6 Marco de trabajo

Un marco o ambiente de trabajo está orientado a la reutilización de componentes permitiendo el desarrollo rápido de aplicaciones. Los marcos de trabajo a emplear para la implementación del sistema son: django 1.6 y bootstrap 3.0.

1.6.1 Django 1.6.1

²⁹ <http://www.desarrolloweb.com/manuales/manual-jquery.html>

³⁰ JQuery: sitio oficial, <http://jquery.com/>

Django es un framework web de código abierto escrito en [PythonPython](#) que permite construir aplicaciones web más rápido y con menos código (2016).

Ventajas de Django (Python):

- Aparte de las ventajas que tiene por ser framework, Django promueve el desarrollo rápido, se construyen aplicaciones en cuestión de días y con el conocimiento suficiente esos días se pueden reducir a horas.
- Impulsa el desarrollo de código limpio al promover buenas prácticas de desarrollo web.
- Django usa una modificación de la arquitectura Modelo-Vista-Controlador (MVC), llamada MTV (Model - Template - View), que sería Modelo-Plantilla-Vista, esta forma de trabajar permite que sea pragmático.

1.6.2 Bootstrap 3.0

Bootstrap es una herramienta para crear interfaces de usuario totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías. Es un soporte bastante bueno (casi completo) con HTML5 y CSS3, permitiendo ser usado de forma muy flexible para desarrollo web con unos excelentes resultados (Fontela, Alvaro).

1.7 Lenguaje de modelado

El lenguaje de modelado es útil debido a que permiten detectar problemas y proponer soluciones mediante el uso de notaciones gráficas. Es esencial en la construcción de software para comunicar la estructura de un sistema complejo, especificar el comportamiento deseado del sistema y descubrir oportunidades de simplificación.

1.7.1 Lenguaje de Modelado Unificado

El lenguaje unificado para la construcción de modelos (UML por sus siglas en inglés) se define como un lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas informáticos. El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño de las clases, hasta la implementación y configuración de los diagramas de despliegue (Booch, y otros).

1.7.2 Notación de Modelado de Procesos de Negocio (BPMN)

En este proyecto, los procesos del negocio se comportan como un flujo de actividades o procesos al ser especificadas claramente en la metodología CRISP-DM las fases que son necesarias para la extracción de conocimiento. Entre estas fases existen dependencias, las salidas de unas constituyen entradas de otras. Por lo que se decidió utilizar BPMN en el actual trabajo.

BPMN provee una notación que es comprendida tanto por usuarios, analistas, grupo de desarrollo que implementan las tecnologías que van a ser usadas en los procesos, como por las personas del negocio que se encargarán de administrar y controlar los procesos. Define un estándar que sirve de puente entre el diseño de los procesos del negocio y los procesos de implementación. Además, brinda la capacidad de comprender los procesos internos de negocio mediante una notación gráfica, dándole a las organizaciones la posibilidad de comunicar esos procesos en un estándar³¹.

1.8 Herramientas de desarrollo

Para el desarrollo de la aplicación se utilizarán un conjunto de tecnologías y herramientas que se describen a continuación:

1.8.1 Herramienta CASE

Dentro de las herramientas claves en el desarrollo de aplicaciones informáticas se encuentran las herramientas de Ingeniería de Software Asistida por Ordenador (CASE³²), las cuales son las encargadas de ayudar en el ciclo de desarrollo, con el fin de aumentar la productividad y reduciendo el costo en términos de tiempo y dinero. En el ciclo de desarrollo pueden ayudar en el proceso de diseño del proyecto, en el cálculo de costos, pueden implementar una parte del código compilación automática y documentación (Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet., 2010).

En el presente trabajo se utilizó Visual Paradigm para UML.

Visual Paradigm 8.0 para UML

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y

³¹ <http://www.bpmn.org/>

³² del inglés: Computer Aided Software Engineering

documentación (Sistema para la integración de datos bibliográficos almacenados en diversos formatos., 2013).

La herramienta CASE Visual Paradigm fue escogida para el desarrollo de la aplicación debido a que utiliza UML como lenguaje de modelado, agiliza la creación de los diferentes diagramas definidos en las metodologías AUP y CRISP-DM, se integra con Python y PostgreSQL, se puede utilizar en varios sistemas operativos, soporta el modelamiento de procesos de negocio con BPMN y genera una excelente documentación en varios formatos (.jpg, .html, .pdf, entre otros.).

1.8.2 Entorno de Desarrollo Integrado

Una de las herramientas que juegan un papel importante en el desarrollo de soluciones informáticas son los Entornos de Desarrollo Integrado (IDE³³). Estos ofrecen facilidades al equipo de desarrollo cuando se implementan las aplicaciones debido a que permite la identificación de errores comunes que se comenten a diario (Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet., 2010).

PyCharm 4.5.4

Se decidió utilizar [PyCharm](#)³⁴ 4.5.4 para el desarrollo de la aplicación. Es un IDE o entorno de desarrollo integrado multiplataforma utilizado para [desarrollar](#) en el lenguaje de programación Python. Proporciona análisis de código y soporte para el desarrollo web con Django. Entre sus principales características se encuentra el autocompletado, el resaltador de sintaxis para código Python, HTML, CSS, JavaScript, así como para las plantillas de Django. PyCharm permite la integración de pruebas unitarias y múltiples opciones para refactorizar el código.

1.8.3 PostgreSQL 9.4

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente (Martínez, 2013). Es usado para manejar grandes cantidades de información. Es multiplataforma, aporta flexibilidad y permite definir funciones personalizadas por medio de varios lenguajes (python, java, php) (Anton, César, 2015).

³³ del inglés: Integrated Development Environment

³⁴ <http://www.prnewswire.com/news-releases/con-pycharm-los-desarrolladores-de-python-finalmente-obtienen-una-ide-poderosa-104948549.html>

Se utiliza PgAdmin en su versión 1.20.0 como herramienta de código abierto con el propósito general de diseñar, mantener y administrar las bases de datos de PostgreSQL.

1.9 Conclusiones parciales

En este capítulo se presentaron determinadas definiciones para una mejor comprensión del tema. Se realizó el estudio de las tendencias actuales de las herramientas que emplean el análisis de comentarios de usuarios para determinar el funcionamiento de cada una y sacar provecho de sus características. Para el desarrollo de la aplicación web se determinó emplear dos metodologías: AUP y CRISP-DM para guiar las actividades del proceso de desarrollo del sistema y orientar el trabajo de la minería de datos respectivamente. Además, se tomó como lenguaje de programación Python en su versión 2.7 por las bibliotecas que contiene y como entorno de desarrollo integrado (IDE) PyCharm en su versión 4.5.4 como soporte para el desarrollo de la aplicación web.

Capítulo 2. Características del sistema

2.1 Introducción

En el presente capítulo se tiene un enfoque práctico y funcional, cuyo objetivo es presentar el resultado del diagnóstico y el análisis de tendencia y estado actual de la minería de opinión aplicado a los comentarios de usuarios. A continuación, se muestra la propuesta de solución al problema planteado. Se abordan los procesos de negocio, diagramas y descripciones asociados al proceso de minería de opinión para la clasificación de comentarios de usuarios. Además, se definen los requisitos funcionales, no funcionales y los actores³⁵ del sistema.

2.2 Propuesta de solución

La solución propuesta está diseñada para ayudar a los desarrolladores a clasificar de forma automática los comentarios de los usuarios que utilizan los softwares, para darle solución a los errores más inmediatos de las aplicaciones de manera eficiente³⁶.

Inicialmente el usuario accede al sitio escribiendo sus credenciales. Una vez autenticado podrá entrar al registro de documentos y envío del mismo para que se guarde en el sistema. Una vez subido el archivo se procesan los datos; este procesamiento lo realiza el sistema de forma no visible al usuario, (se pone el estado del documento como pendiente y cuando se procesa cambia a procesado). Después el usuario podrá agregar más documentos o seleccionar uno de los que se procesó para ver los resultados, editarlo o eliminarlo. Si se desea ver los resultados se mostrará una tabla con los comentarios de los usuarios clasificados en positivo, negativo y neutro. Además, se mostrarán por categorías los comentarios con sus respectivas clasificaciones y sus gráficas. Una vez realizada la clasificación los usuarios tienen la posibilidad de exportar los resultados con extensión .pdf. En la figura 4, se muestra la propuesta de solución.

Una de las etapas más necesarias en esta propuesta es el procesamiento de los comentarios de usuarios. En este proceso se preparan y limpian los datos (eliminar signos de puntuación, palabras vacías, números, convertir las palabras a minúscula, llevarlas a su raíz y corregirlas ortográficamente), se convierten en una estructura de datos intermedia (vector de sentencias) y se le aplica el algoritmo de clasificación.

³⁵ Un rol jugado por un usuario o cualquier otro sistema que interactúa con el sujeto.

³⁶ Es la capacidad de disponer de alguien o de algo para conseguir un efecto determinado.



Figura 4. Propuesta de solución

2.3 Modelo de negocio

A continuación, se describen los procesos de negocio utilizando la notación BPMN. Por su importancia, sólo se muestran las descripciones correspondientes a los procesos: General, Preparar datos, Aplicar minería y Evaluar comportamiento.

2.3.1 Proceso general

En la tabla 1, se muestra la descripción del “Proceso general”.

Ficha de Proceso	
Proceso	Proceso general
Entradas	Fuentes de donde proceden los datos (los comentarios de los usuarios de aplicaciones de software que se encuentran en un documento .txt) a preparar. Para luego filtrarlos y crear las vistas minables.
Salidas	Patrones de clasificación de comentarios de usuarios en forma de conocimiento.
Descripción del Proceso	
La ejecución de la herramienta se ha dividido en 4 subprocessos donde se evidencian las fases del flujo de KDD. Se preparan los datos y luego se aplica el algoritmo de Minería de Datos. Posteriormente se	

evalúa el comportamiento obtenido, verificando si los resultados son aceptados, en caso de serlo, se muestran los resultados. Si los resultados no son aceptados, se necesitará nuevamente preparar o los datos, si no es necesario aplicar el proceso de preparar los datos, entonces se aplica nuevamente la minería.

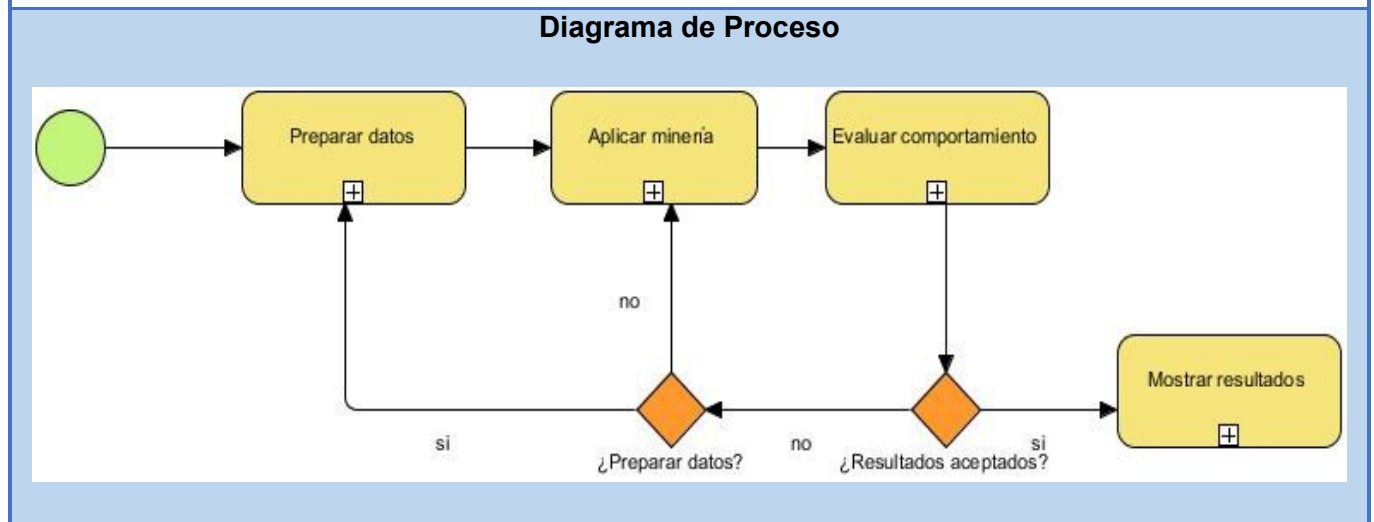


Tabla 1. Descripción del “Proceso general”

2.3.2 Preparar datos

En la tabla 2, se muestra la descripción del proceso “Preparar datos”.

Ficha de Proceso	
Proceso	Preparar datos
Entradas	Comentarios de los usuarios de aplicaciones de software que se encuentran en un documento .txt.
Salidas	Diccionario con los comentarios de usuarios listos para aplicar minería.
Descripción del Proceso	
<p>La preparación de los datos se ha dividido en 4 subprocesos acordes con las tareas que se realizan en esta fase de KDD. Primero se obtienen las opiniones de los usuarios del documento .txt, para luego procesarlas y transformarlas (dividir las sentencias de cada comentario de usuario, después separarlas en palabras, quitar los signos de puntuación, eliminar las palabras que no contienen información útil, llevar las palabras a la raíz y corregirlas ortográficamente). Finalmente se mezclan para obtener los datos listos para su clasificación.</p>	

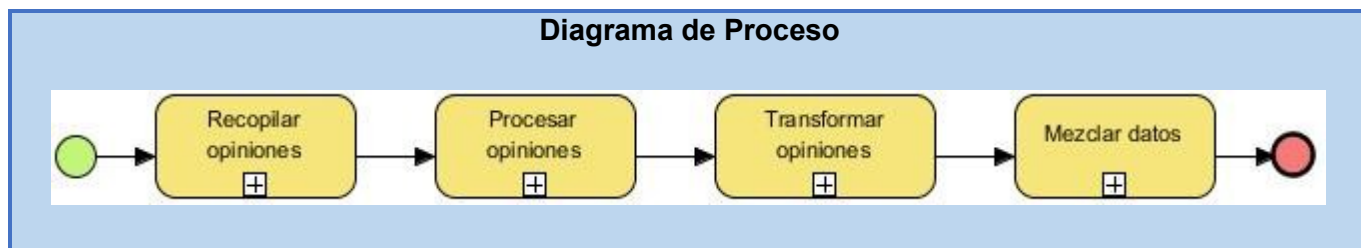


Tabla 2. Descripción del proceso “Preparar datos”

2.3.3 Aplicar minería

En la tabla 3, se muestra la descripción del proceso “Aplicar minería”.

Ficha de Proceso	
Proceso	Aplicar minería
Entradas	Parámetros del algoritmo y el diccionario con los comentarios de usuarios listos para aplicar minería.
Salidas	Comentarios de usuarios clasificados.
Descripción del Proceso	
Los parámetros del algoritmo (clasificación) y el diccionario con los comentarios se envían a la biblioteca de algoritmos de Minería de Datos, la cual ejecuta un clasificador bayesiano que permita obtener los comentarios de usuarios clasificados, guardándose en la base de datos.	
Diagrama de Proceso	

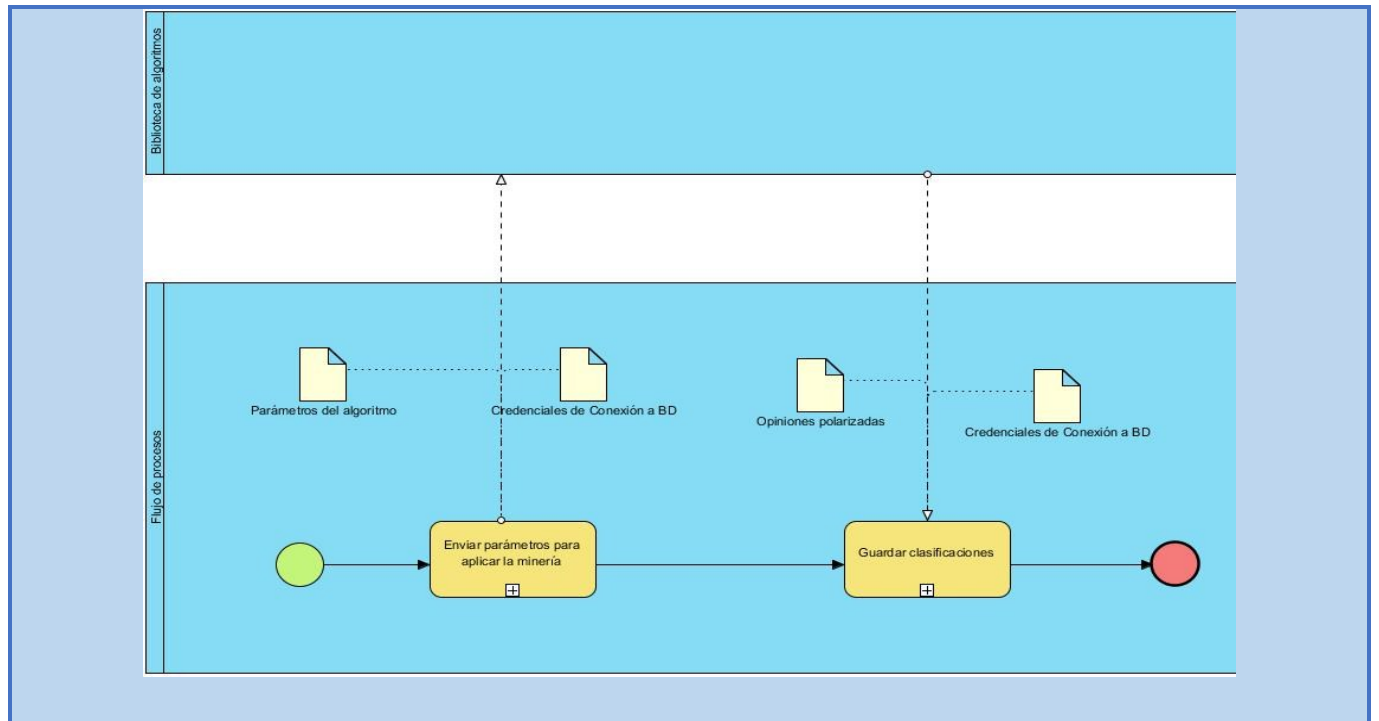


Tabla 3. Descripción del proceso “Aplicar minería”

2.3.4 Evaluar comportamiento

En la tabla 4, se muestra la descripción del proceso “**Evaluar comportamiento**”.

Ficha de Proceso	
Proceso	Evaluar comportamiento.
Entradas	Credenciales de conexión de la base de datos.
Salidas	Patrones de clasificación de comentarios de usuarios en forma de conocimiento.
Descripción del Proceso	
Se obtiene los comentarios de usuarios clasificados en positivo, negativo y neutro previamente guardados en la base de datos y se crean gráficas que expresan los patrones de clasificación de comentarios de usuarios en forma de conocimiento. Las clases obtenidas se utilizan como base para la clasificación de nuevos comentarios	
Diagrama de Proceso	

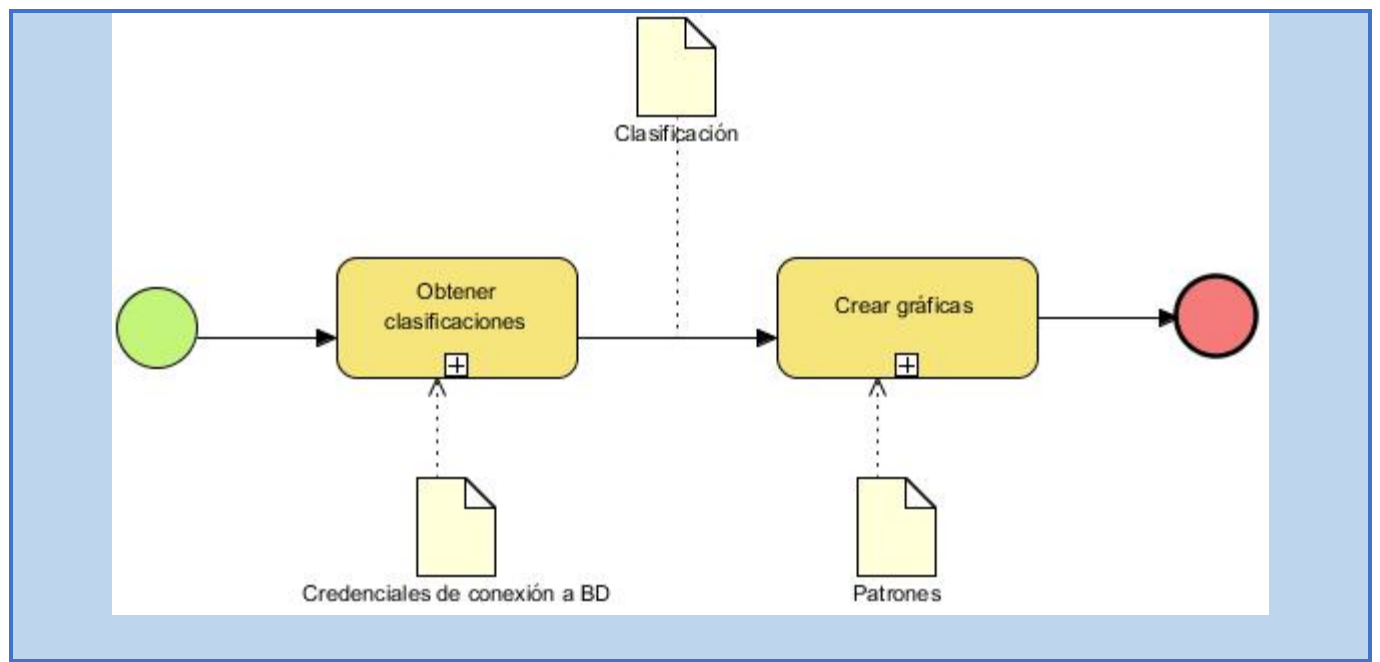


Tabla 4. Descripción del proceso “Evaluar comportamiento”

2.4 Levantamiento de requisitos

Los requisitos de software son las cualidades y características que el producto debe tener. Estos se dividen en requisitos funcionales y requisitos no funcionales (Pressman, 2000). Mediante la comunicación con el cliente los mismos quedaron definidos de la siguiente manera.

2.4.1 Requisitos funcionales del sistema

Los requisitos funcionales son las condiciones o capacidades que debe cumplir el sistema (Pressman, 2000).

A continuación, se muestran los requisitos funcionales que el sistema debe permitir:

R1: **Autenticar usuario:** Identifica a los usuarios y los autoriza a entrar en el sistema.

R2: **Registrar usuario:** Permite a los usuarios registrarse en la aplicación.

R3: **Gestionar documento:** Permite a los usuarios crear, editar, eliminar y observar los documentos que ha subido.

R3.1: **Crear documento:** Se guarda el archivo con los comentarios de usuarios en el sistema.

R3.2: **Editar documento:** Edita el documento que el usuario subió en el sitio.

R3.3: **Eliminar documento:** Elimina el documento del registro de documentos.

R3.4: **Mostrar documento:** Muestra los documentos que ha subido los usuarios.

R4: **Aplicar minería:** Se procesa el archivo subido por el usuario para obtener los comentarios clasificados en positivos, negativos y neutros.

R4.1: **Integración y recopilación de los datos:** Se extraen los comentarios de usuarios de un documento local.

R4.2: **Preparación de los datos:** Selecciona, limpia y transforma los datos recopilados.

R4.3: **Aplicar el modelo de clasificación:** Obtiene los datos preparados y se aplica el algoritmo de clasificación.

R4.4: **Evaluar comportamiento:** Evaluar los resultados que arroja el clasificador.

R5: **Mostrar resultados:** Se observan los comentarios de un documento clasificados en positivo, negativo y neutro y las categorías a las que pertenecen.

R5.1: **Graficar categorías:** Muestra gráficamente según las categorías los comentarios positivos y negativos.

R5.2: **Graficar comentarios:** Muestra gráficamente los comentarios positivos, negativos y neutros.

R5.3: **Listar categorías:** Se muestran las categorías de los comentarios y se observan los comentarios de un documento clasificados en positivo, negativo y neutro.

R6: **Exportar resultados:** Se exportan los resultados obtenidos con extensión pdf.

En la tabla 5, se muestra la especificación del RF#1 “Autenticar usuario”. La descripción de los demás requisitos con más detalles se puede consultar en el Anexo 2 Especificación de los requisitos funcionales.

Precondiciones	1. El usuario se encuentra registrado en el sistema.
Flujo de eventos	
Flujo básico	
1. Entrar a la página de inicio	
2. Presionar el botón “Autenticarse”.	
3. Insertar los datos de los campos: Usuario y Contraseña.	
4. Confirmar los datos escritos haciendo clic en el botón “Entrar”.	
Pos-condiciones	
1. El usuario accede a la aplicación.	
Validaciones	
1. Se debe escribir correctamente los campos del usuario y la contraseña.	
Pos-condiciones	
1. El usuario no se encuentra registrado en el Sistema.	
2. Ir a la descripción del RF#1: “Registrar usuario”.	
Prototipo de Interfaz.	

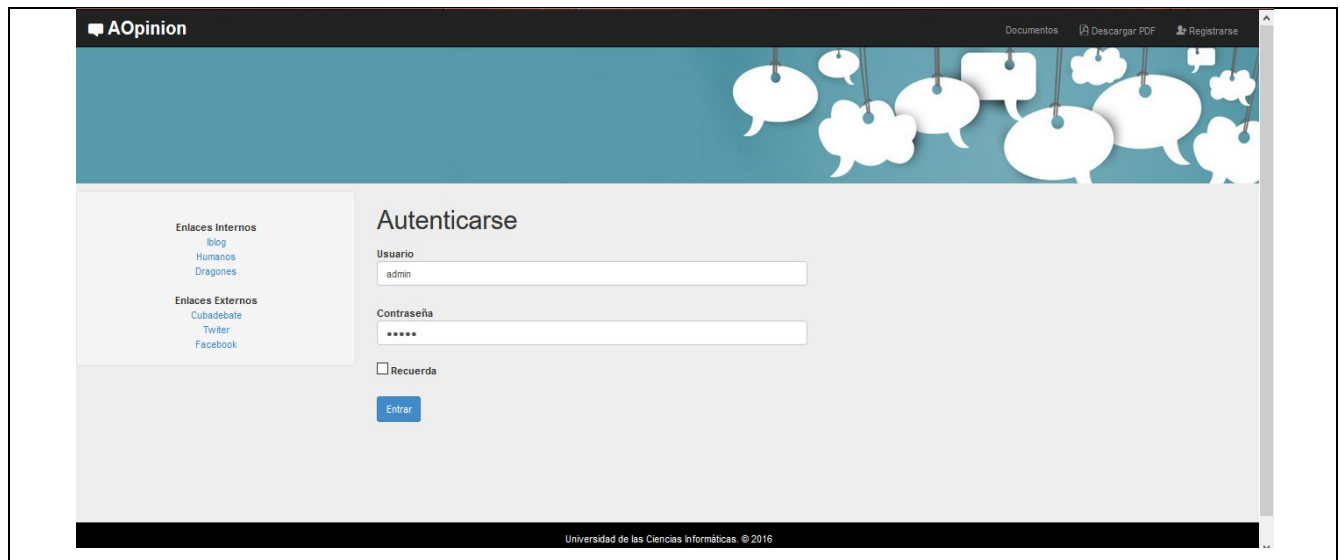


Tabla 5. Especificación del RF#1 “Autenticar usuario”

2.4.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable (Pressman, 2000).

Usabilidad

- Se requiere un nivel medio de conocimientos de informática, pues la aplicación contará con una ayuda para guiar al usuario.

Apariencia

- Todos los textos y mensajes en pantalla aparecerán en idioma español.
- La interfaz de aplicación está diseñada con un estilo sencillo y de fácil manejo con la intención de que el cliente interactúe con el sistema.

Seguridad

- El usuario para entrar al sistema será autenticado mediante su nombre de usuario y su contraseña, la cual se encuentra cifrada en la base de datos utilizando el algoritmo MD5.

Software

Para la PC cliente:

- La aplicación correrá en sistemas con Microsoft Windows XP (arquitecturas x86 y x86-64) o superior, así como GNU/Linux (arquitecturas x86 y x86-64).
- Se requiere el uso de un navegador web.

Para el servidor:

- Se requiere el gestor de base de datos PostgreSQL 9.4.
- Python 2.7.
- Servidor de aplicaciones Nginx, Gunicorn o Gevent.
- Las bibliotecas como NLTK, TextBlob, psychopg 2.
- La aplicación correrá en sistemas con Microsoft Windows XP (arquitecturas x86 y x86-64) o superior, así como GNU/Linux (arquitecturas x86 y x86-64).

Hardware:

Para la máquina cliente:

Requerimientos recomendados:

- 512 MB de RAM o más.
- Un procesador de 1.0 GHz o superior.
- Tarjeta de red para establecer la conexión.

Para la máquina servidor:

Requerimientos recomendados:

- 4 GB de RAM.
- 40 GB de Disco Duro(HDD).
- Dos núcleos, cada uno de 2.0 GHz o superior.
- Tarjeta de red para establecer la conexión.

2.5 Modelo de casos de uso del sistema

2.5.1 Definición de los actores

Los actores del sistema son personas u otros sistemas que interactúan con el sistema. Incluyen a operadores humanos, sistemas externos y entidades abstractas como el tiempo (Larman, 1999).

En la tabla 6, se muestra el actor del sistema y su descripción.

Actor	Descripción
Usuario	Es la persona que puede realizar una serie de operaciones, luego de autenticarse.

Tabla 6. Actor del sistema

2.5.2 Diagrama de casos de uso del sistema

Un caso de uso es una secuencia de interacciones entre un sistema y alguien o algo que usa alguno de sus servicios. Permite definir los límites del sistema y el entorno. Son descripciones de las funcionalidades del sistema independientes de la implementación.

En la figura 5, se muestra el diagrama de casos de uso del sistema.

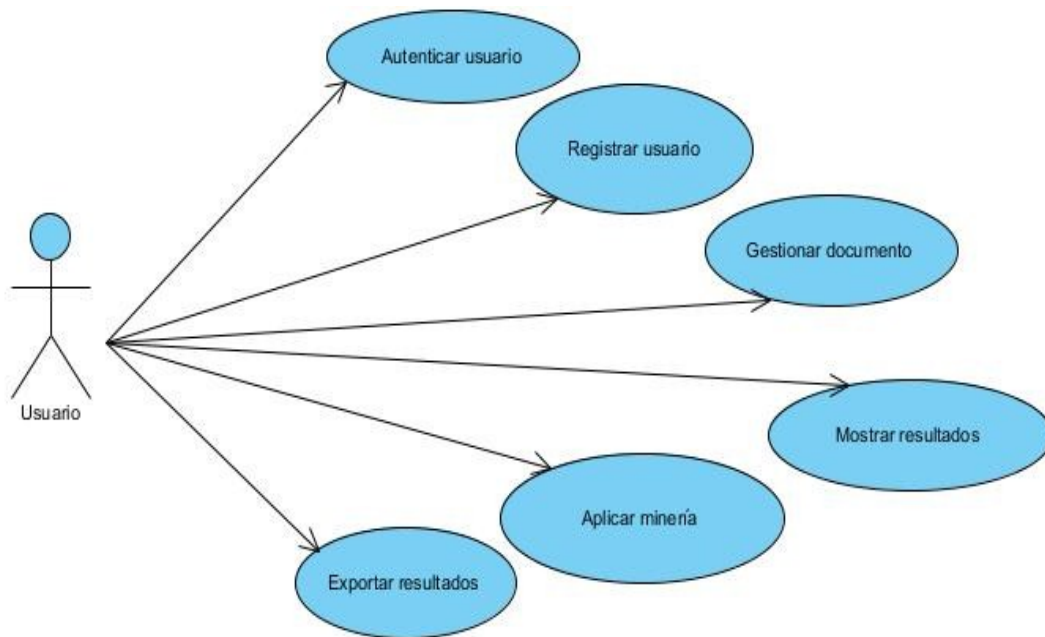


Figura 5. Diagrama de casos de uso del sistema

2.6 Conclusiones parciales

Se realizó un análisis de los procesos fundamentales que intervienen en la extracción de conocimiento para la obtención de la clasificación de las opiniones de los usuarios. Dichos procesos fueron modelados con la notación BPMN, permitiendo una clara comprensión de los mismos. Se identificaron los requerimientos a cumplir por parte del sistema, tanto funcional como no funcional, aportando la

información necesaria para la creación y despliegue de la aplicación. Se diseñó el modelo de casos de uso del sistema que constituye una entrada importante para representar los diferentes flujos de trabajo.

Capítulo 3. Diseño del sistema

3.1 Introducción

El diseño del sistema requiere de pasos necesarios para que se produzcan los resultados esperados. A continuación, en el presente capítulo se define el modelo físico de los datos, el diagrama de paquetes y los diagramas de clases del diseño con sus relaciones. Se brinda una vista general de la arquitectura y los patrones de diseño utilizados en la implementación de la aplicación AOpinion.

3.2 Diagrama de paquetes

El diagrama de paquetes del diseño muestra cómo un sistema está dividido en agrupaciones lógicas y las dependencias entre esas agrupaciones (Larman, 1999).

El diagrama de paquetes se estructuró en concordancia con la programación del sistema:

- **App**: maneja las configuraciones y las rutas que están disponibles en el proyecto.
- **AOpinion**: maneja la lógica del negocio, que contiene las funciones de la vista y las clases del modelo.
- **Templates**: contiene todas las plantillas del sistema.
- **Biblioteca**: contiene las bibliotecas de python.
- **Django**: contiene los archivos del framework de desarrollo.

En la figura 6, se muestra el diagrama de paquetes.

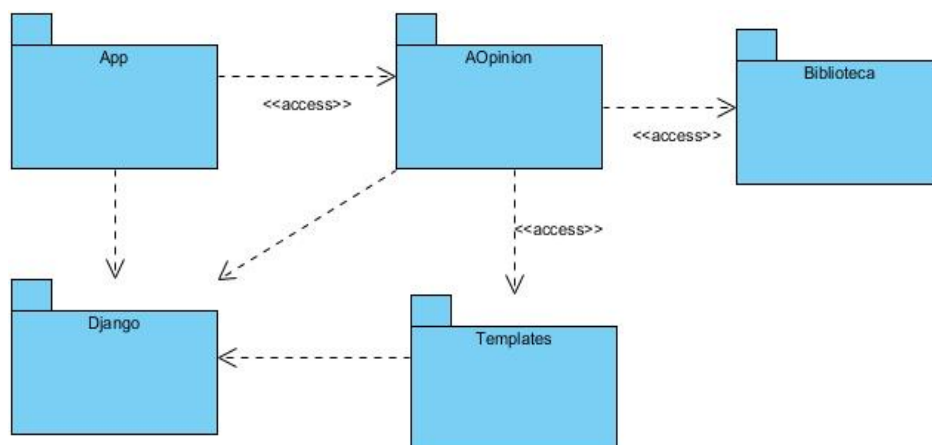


Figura 6. Diagrama de paquetes

3.3 Diagrama de clases de diseño

Los diagramas de clases especifican la estructura de clases de un sistema, así como sus relaciones. Definen las relaciones de dependencia, generalización y asociación de clases que constituyen el sistema.

En la figura 7, se muestra el diagrama de clases del diseño del sistema, utilizando estereotipos web, correspondiente al CU: "Autenticar usuario". La descripción de los demás se pueden consultar en el Anexo #2 Diagramas de clases del diseño.

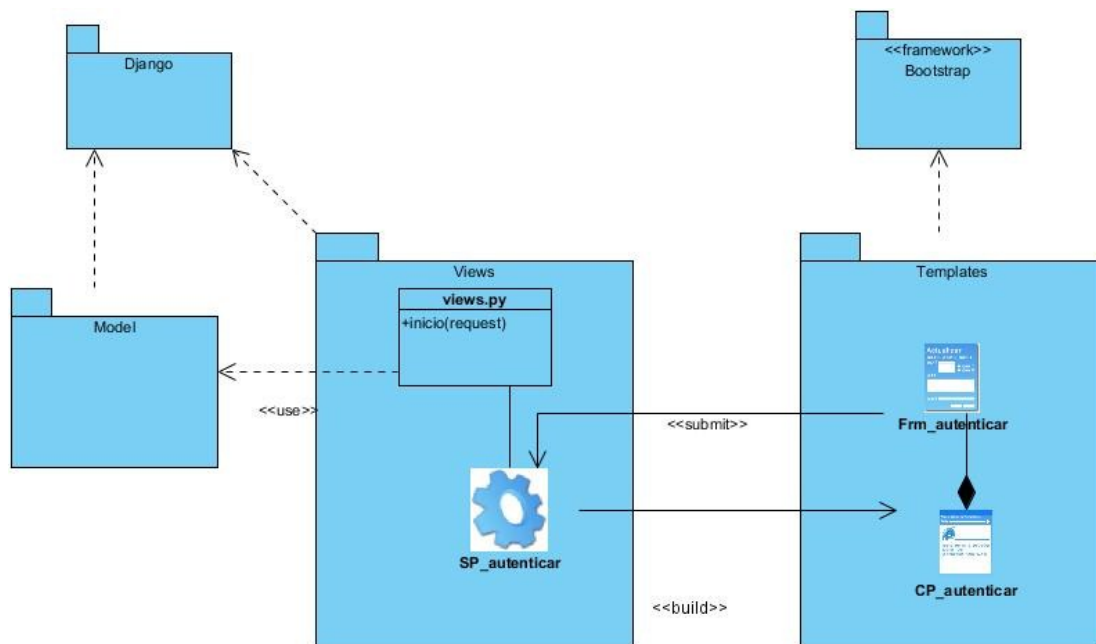


Figura 7. Diagrama de clases del diseño, CU: "Autenticar usuario"

3.4 Modelo físico de los datos

Describe las representaciones físicas de los datos utilizados en la aplicación, los cuales serán almacenados en base de datos. Los elementos esenciales del diagrama son las entidades, los atributos y las relaciones entre las entidades.

- Las **entidades** son objetos de los que el sistema necesita guardar información.
- Los **atributos** son las características asociadas a una entidad, pueden clasificarse en obligatorios, opcionales, claves foráneas o claves primarias.
- Las **relaciones** muestran la forma en que dos entidades se asocian. Se representan mediante una línea que une a las dos entidades implicadas.

En la figura 8, se muestra el modelo físico de la base de datos.

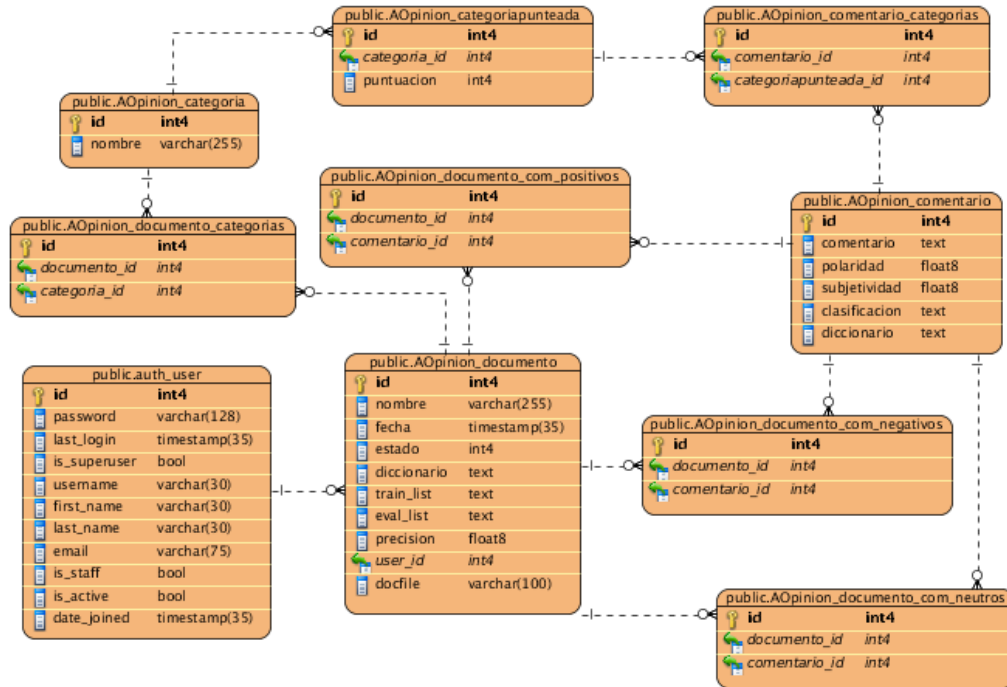


Figura 8. Modelo físico de la base de datos

3.5 Diseño de la arquitectura

La arquitectura de software ~~La arquitectura de software~~³⁷ es un conjunto de patrones que proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software ~~programadores, analistas y todo el conjunto de desarrolladores del software~~ compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación.

Para el desarrollo del sitio se empleará la arquitectura cliente-servidor donde cada usuario autorizado a acceder a la aplicación AOpinion podrá realizar peticiones al servidor donde se encontrará alojado el sistema.

En la figura 9, se muestra arquitectura cliente-servidor.

³⁷ http://www.ecured.cu/Arquitectura_de_software.

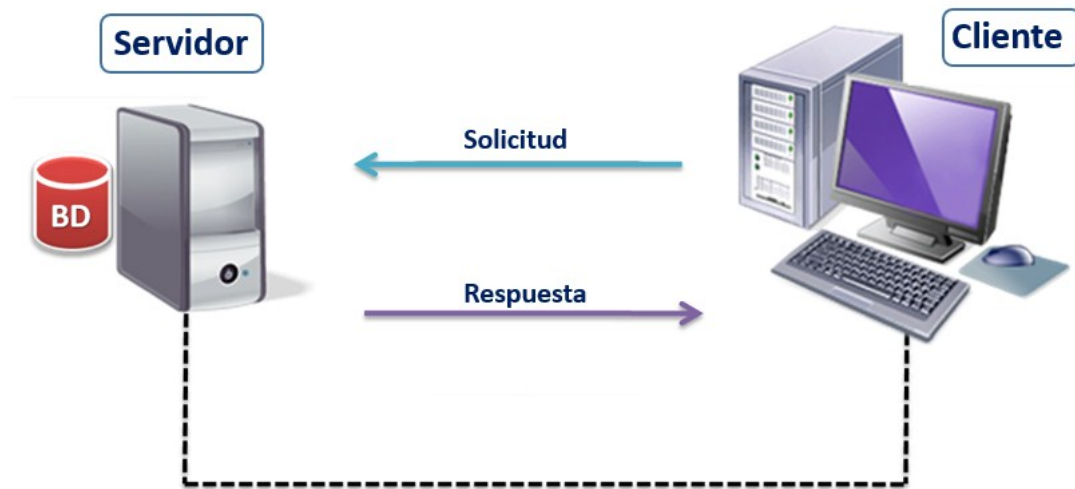


Figura 9. Arquitectura cliente-servidor

3.6 Patrón arquitectónico

Los patrones³⁸ arquitectónicos son los que definen la estructura de un software, los cuales a su vez se componen de subsistemas con sus responsabilidades. También tienen una serie de directivas para organizar los componentes del sistema, con el objetivo de facilitar la tarea del diseño del mismo.

En la arquitectura Modelo-Vista-Controlador (MVC), el marco de desarrollo Django usa una modificación llamada Model-Template-View (MTV) que sería Modelo- Plantilla-Vista empleado en la implementación de la solución. En Django el modelo sigue siendo modelo, la vista pasa hacer la plantilla y el controlador la vista.

- Modelo: define los datos almacenados, además está en forma de clases de Python. Cada tipo de datos que son almacenados se encuentran en una variable con ciertos parámetros. Esto permite indicar y controlar el comportamiento de los datos.
- Plantilla: es la capa de presentación, básicamente una página HTML con algunas etiquetas extras propias de Django, en sí no solo crea contenido en HTML, también XML, CSS, Javascript, CSV. Recibe los datos de la vista y luego los organiza para la presentación al navegador web.
- Vista: se presenta en forma de funciones en Python. Su propósito es determinar qué datos serán visualizados. Se encarga de tareas como el envío de correo electrónico, la autenticación con

³⁸ http://librosweb.es/libro/django_1_0/capitulo_5/el_patron_de_diseno_mtv.html.

servicios [externosexternos](#) y la validación de datos a través de formularios (**Infante Montero, 2012**).

Este patrón comienza su funcionamiento cuando el navegador envía una solicitud a la vista. La vista interactúa con la modelo para obtener los datos. La vista llama a la plantilla y la plantilla renderiza la respuesta a la solicitud del navegador.

En la figura 10, se muestra el patrón arquitectónico Modelo-Plantilla-Vista.

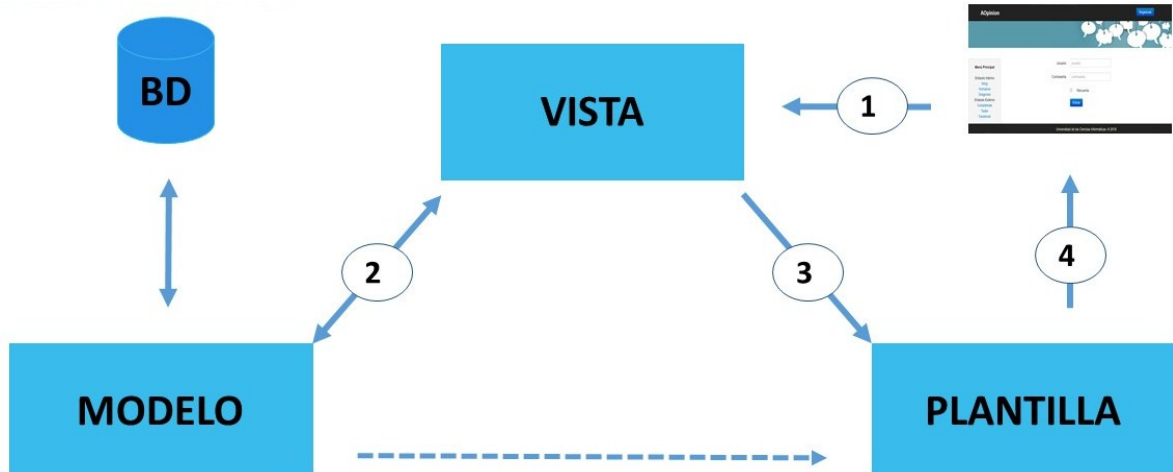


Figura 10. Patrón arquitectónico Modelo- Plantilla-Vista

3.7 Patrones de diseño

Un patrón de diseño es una buena práctica documentada de la solución de un problema que ha sido aplicado satisfactoriamente en múltiples entornos. Es una solución recurrente a un problema común observado o descubierto durante el estudio o construcción de numerosos softwares. Su principal objetivo es incrementar la calidad del software en términos de reusabilidad, mantenimiento y extensibilidad (Larman, 1999).

El Patrón **Decorador** es aplicado a la generación de vistas. Añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia. La solución que ofrece dicho patrón es la de añadir funcionalidad adicional a las plantillas.

- Ejemplo: la utilización de este patrón se evidencia mediante el archivo base.html que almacena el código HTML que es común para todas las vistas del sistema.

En la figura 11, se muestra un ejemplo de la utilización del patrón decorador.

```
{% extends 'base.html' %}
```

Figura 11. Ejemplo de utilización del patrón Decorador

El Patrón **Alta Cohesión** se basa en asignar una responsabilidad de modo que la cohesión siga siendo alta. Es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realizan un trabajo enorme (Larman, 1999).

- Ejemplo: A la clase “CategoriaPunteada” se le asigna responsabilidades con el objetivo de que trabaje en la misma área de la aplicación y que no posea mucha complejidad.

En la figura 12, se muestra un ejemplo de la utilización del patrón Alta Cohesión.

```
class CategoriaPunteada(models.Model):  
    categoria = models.ForeignKey(Categoria)  
    puntuacion = models.IntegerField()
```

Figura 12. Ejemplo de utilización del patrón Alta Cohesión

El Patrón **Bajo Acoplamiento** consiste en asignar una responsabilidad. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con qué las conoce y con qué recurre a ellas (Larman, 1999). Este patrón responde siempre a la necesidad de conservar un mecanismo poco dependiente entre las clases y objetos, de forma tal que pueda reducirse el impacto de los cambios y se incremente la reutilización.

- Ejemplo: Al método “mostrar_documentos” se le asigna responsabilidades de forma que, al comunicarse con la plantilla, en caso de ocurrir algún cambio no afecte a la misma.

En la figura 13, se muestra un ejemplo de la utilización del patrón Bajo Acoplamiento.

```
def mostrar_documentos(request):  
    documentos = Documento.objects.filter(user=request.user).order_by('-fecha')  
    return render(request, 'documento/mostrar_documentos.html', {'documentos':documentos})
```

Figura 13. Ejemplo de utilización del patrón Bajo Acoplamiento

El Patrón **Creador** guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación.

- Ejemplo: el formulario “UserCreationForm” es el responsable de la creación de objetos.

En la figura 14, se muestra un ejemplo de la utilización del patrón Creador.

```
def registrar(request):
    """
    Método para registrar un nuevo usuario
    """
    if request.method == 'POST':
        form = UserCreationForm(request.POST)
        if form.is_valid():
            usuario = User.objects.create_user(username=request.POST['username'], email=request.POST['email'],
                                               password=request.POST['password1'], first_name=request.POST['first_name'])
            usuario.save()
            if usuario:
                return HttpResponseRedirect('/')
        else:
            mensaje = 'El usuario ya esta registrado'
            clase_mensaje = 'success'
            return render_to_response('registrar.html', {'mensaje': mensaje, 'clase_mensaje': clase_mensaje},
                                     context_instance=RequestContext(request))

    return render_to_response('registrar.html', context_instance=RequestContext(request))
```

Figura 14. Ejemplo de utilización del patrón Creador

3.8 Conclusiones parciales

En el presente capítulo se generaron los diferentes diagramas para ayudar y guiar el diseño de la aplicación AOpinion. Fueron explicados los diferentes patrones de diseño a utilizar como buenas prácticas, ofreciendo mayor robustez a la arquitectura MVT. De esta forma quedaron planteados los elementos esenciales del diseño del sistema.

Capítulo 4. Implementación y prueba

4.1 Introducción

Para el desarrollo de la aplicación es de vital importancia la fase de implementación y prueba del sitio AOpinion. A continuación, en el presente capítulo se definen los elementos físicos y de validación del sistema. Se explican los estándares de codificación usados en el código del sitio con el objetivo de definir la escritura y organización del código fuente de la aplicación. Se plasman los resultados de las pruebas realizadas al sistema, con el objetivo de garantizar la correcta ejecución, revisión y retroalimentación³⁹ de las funcionalidades.

4.2 Diagrama de componente

Los diagramas de componentes permiten describir los elementos físicos que integran el sistema y las relaciones que existen entre ellos. “Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente se refiere a los servicios ofrecidos por otro componente (Roldan Diego, 2009).”

En la figura 15, se muestra el diagrama de componentes asociado al sistema implementado y la descripción de los componentes. Se utilizó la dependencia <<use>>.

- **models.py:** declara las clases del modelo.
- **views.py:** declara las funciones de la vista.
- **url.py:** contiene las rutas que están disponibles en el proyecto.
- **cladificador.py:** componente encargado de procesar y clasificar los comentarios de usuarios.
- **templates:** contiene las plantillas del sistema.
- **django:** contiene las clases propias del framework.

³⁹ Es un mecanismo por el cual una cierta proporción de la salida de un sistema se redirige a la entrada, con objeto de controlar su comportamiento.

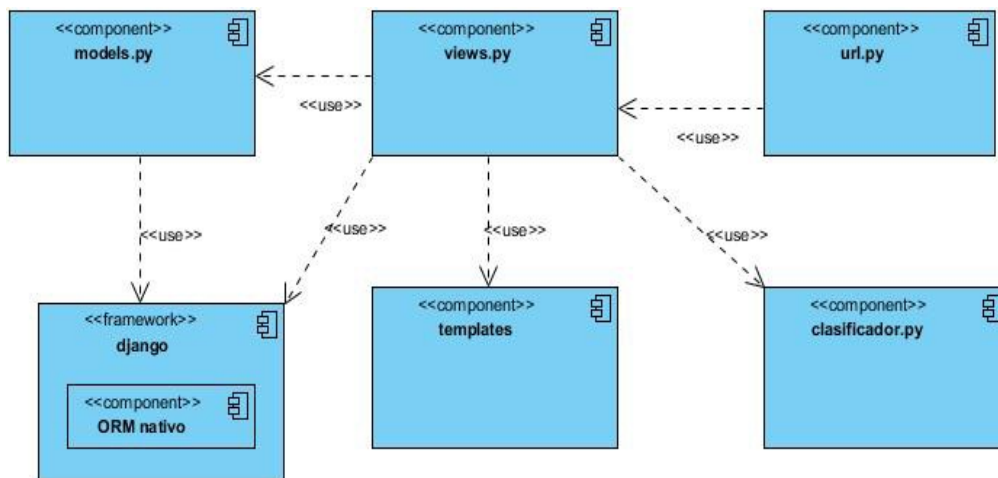


Figura 15. Diagrama de componente

4.3 Estándares de codificación

Con vistas a mejorar el entendimiento del código del sistema para los desarrolladores y alcanzar una uniformidad en el mismo, a continuación, se muestran algunos estándares de codificación utilizados en la implementación.

- **Indentación:**

Se considera que indentar correctamente las instrucciones del programa es una buena práctica para hacer más legible el código. En Python esto no es opcional, dependiendo de la posición en que una instrucción concreta esté colocada, el intérprete realizará una tarea u otra o incluso considerará que la aplicación no está correctamente construida y mostrará un error o ni siquiera la ejecutará. En el caso de Python, la indentación es obligatoria, pues de ella dependerá su estructura. La indentación se puede realizar añadiendo espacios o tabulaciones. En la solución desarrollada se utilizó la indentación a cuatro espacios (tabulación).

- **Variables:**

Los nombres de las variables se escriben en minúscula con las palabras separadas con guiones bajos, tantos como sea necesario para mejorar la legibilidad.

- **Comentarios, separadores, líneas, espacios en blanco y márgenes:**

Los comentarios de bloque se aplican al código que se encuentra a continuación, y se indentan al mismo nivel que dicho código. Cada línea de un comentario de bloque comienza y termina con tres comillas dobles. Un comentario en línea es un comentario que se encuentra en la misma línea que una sentencia. Los comentarios en línea se separan por al menos dos espacios de la sentencia que comentan. Comienzan con el símbolo # y un espacio. Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

- **Clases y objetos:**

El nombre empleado para las clases, objetos, atributos y funciones deben ser entendibles. Los nombres de las clases deben comenzar con la primera letra en mayúscula, se debe utilizar la convención CapWords. Ejemplo: class CategoriaPunteada(models.Model).

El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula y estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto, la segunda palabra de igual forma comenzará con minúscula. Para nombrar las funciones se utilizó la convención mixedCase (la primera palabra será con minúscula). Ejemplo: def inicio (request).

4.4 Diagrama de despliegue

El diagrama de despliegue contiene los nodos que forman la topología hardware, los softwares necesarios para su funcionamiento y los protocolos de comunicación sobre la que se ejecuta el sitio AOpinion.

Nodos del diagrama de despliegue.

- **PC_Cliente-Navegador Web:** representa las PC de los clientes mediante las que se accederá al sistema disponible en el servidor de aplicaciones.
- **Servidor de Aplicaciones Web Django:** es el servidor donde estará disponible el sistema, además debe estar instalado el lenguaje de programación Python 2.7, las bibliotecas (NLTK, TextBlob y Psycopg 2) y en dependencia del servidor de aplicaciones utilizado para el despliegue Nginx, Gunicorn o Gevent.
- **Servidor de Bases de Datos PostgreSQL:** es el servidor de base de datos que guarda todas las entidades creadas por el proyecto.

Se utilizaron los protocolos de comunicación TCP/IP y HTTP debido a que la aplicación se desarrolla sobre tecnología web.

- **TCP/IP:** se utiliza para enlazar las computadoras que usan sistemas operativos similares o diferentes, en redes de área local (LAN). En el sistema AOpinion se utiliza para interconectarlo con el servidor de base de datos.
- **HTTP:** protocolo de transferencia de hipertexto, es usado en cada transacción de la web. Este protocolo define la sintaxis y semántica que utilizan los elementos software de la arquitectura web (cliente, servidor, proxy para comunicarse). En el sistema se utiliza para el acceso de los clientes web a los servicios que brinda la aplicación.

En la figura 16, se muestra el diagrama de despliegue.

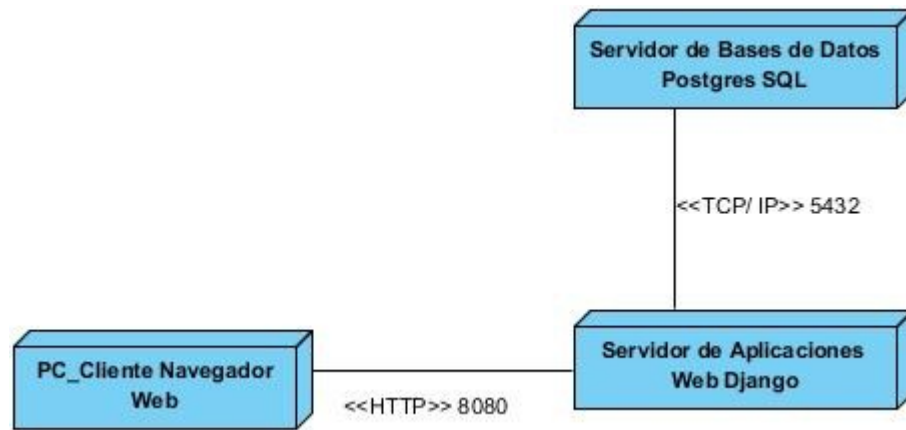


Figura 16. Diagrama de despliegue

4.5 Pruebas

Las pruebas de software son actividades con las cuales el sistema o parte de él es ejecutado bajo condiciones específicas. Los resultados obtenidos permiten realizar una evaluación del mismo con el objetivo de valorar su calidad. Las pruebas son imprescindibles para garantizar la calidad de la aplicación representando una revisión final de las especificaciones del diseño y del código (Proceso de pruebas para productos de software en un laboratorio de calidad, 2014). En el presente trabajo se realizarán pruebas al software y se evaluará el algoritmo de clasificación empleado.

4.5.1 Estrategia de prueba

Una estrategia de prueba del software integra los métodos de diseño de caso de pruebas del software en una serie bien planeada de pasos que desembocará en la eficaz construcción del software. La estrategia proporciona un mapa que describe los pasos que se darán como parte de la prueba, indica cuando se planean y cuando se dan estos pasos, además de cuanto esfuerzo, tiempo y recursos consumirán. Por tanto, cualquier estrategia de prueba debe incorporar la planeación de pruebas, el diseño de casos de pruebas, la ejecución de pruebas y la recolección y evaluación de los datos resultantes (Pressman, 2010).

4.5.2 Métodos de pruebas

Los métodos de pruebas definen estrategias para descubrir fallos en el sistema (**Pressman, 2005**)

4.5.2.1 Pruebas de caja blanca

Las pruebas de Caja Blanca se nombran de esta forma porque a diferencia de las pruebas de Caja Negra que actúan sobre la interfaz, estas revisan la parte interna del software, específicamente sobre el

código fuente. Se basan en el examen minucioso de los detalles procedimentales. Se comprueban los caminos lógicos del sistema generando casos de prueba que ejerciten las estructuras condicionales y los bucles. Es por esto que las Pruebas Unitarias se basan en las Técnicas de Pruebas de Caja Blanca (Pressman, 2005).

Las pruebas unitarias permiten probar, como su nombre lo indica, cada unidad independiente del software. Actúan esencialmente sobre el código fuente y sobre los elementos básicos de la interfaz de la aplicación (Pressman, 2005).

Se utilizó la biblioteca de Python (unittest), las pruebas fueron hechas dentro del archivo test.py, creado por el marco de trabajo Django. Las pruebas fueron realizadas de forma automática haciendo uso del sistema de prueba que posee Django y para ejecutar `los-test` o iniciar el servidor de prueba se hace uso del comando “python manage.py test”.

En la figura 17, se muestra el resultado de las pruebas realizadas.

```
class Test_models(unittest.TestCase):

    def test_Categoria(self):
        test_categoria = False
        try:
            categoria = Categoria(nombre='software')
            categoria.save()
            test_categoria = True
        except:
            pass
        self.assertTrue(test_categoria)

    def test_CategoriaPunteada(self):
        test_categoriaPunteada = False
        try:
            categoriaPunteada = CategoriaPunteada(categoria='software', puntuacion='3')
            categoriaPunteada.save()
            test_categoriaPunteada = True
        except:
            test_categoriaPunteada = True
            pass
        self.assertTrue(test_categoriaPunteada)

Terminal

Creating test database for alias 'default'...
..
-----
Ran 2 tests in 0.007s

OK
Destroying test database for alias 'default'...
```

Figura 17. Resultado de las pruebas realizadas

4.5.2.2 Pruebas de caja negra

Las pruebas de caja negra “se centran en los requisitos funcionales del software, tiene como objetivo verificar que la entrada se acepta correctamente y que ejerciten en su totalidad todos los requisitos funcionales de un sistema” (Pressman, 2006).

Son pruebas funcionales sin acceso al código fuente de las aplicaciones, y se llevan a cabo sin tener conocimiento de la estructura y funcionamiento interno del sistema. La prueba de caja negra intenta encontrar las siguientes categorías de errores (Pressman, 2006):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externos.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

4.5.3 Casos de prueba

“Un caso de prueba es un conjunto de acciones con resultados y salidas previstas basadas en los requisitos de especificación del sistema”. (Aristegui O, 2010).

En la tabla 7, se muestra el caso de prueba aplicado al RF#1: “Autenticar usuario”, utilizando la técnica partición equivalente. Los demás casos de pruebas se pueden consultar en el Anexo #4 Casos de prueba de los requisitos funcionales.

Caso de prueba RF1			
Condiciones de ejecución: Autenticar usuario			
Claves válidas	Claves no válidas	Resultado esperado	Resultado de prueba
El usuario escribe su usuario y contraseña.		El usuario entra al sistema mostrando un registro con los documentos que ha subido.	Satisfactoria
	El usuario escribe mal su usuario o contraseña.	El usuario no accede al sistema y se muestra un mensaje “Error en sus datos”.	Satisfactoria

Tabla 7. Caso de prueba del RF#1: “Autenticar usuario”

4.6 Resultados de las pruebas

El sistema se sometió a una serie de pruebas de validación que permitieron comprobar el correcto funcionamiento de los requisitos implementados mediante la utilización del método de Caja Negra. La técnica empleada fue la de partición equivalente.

En la tabla 8, se muestran los resultados de las pruebas.

Pruebas realizadas	Iteración	No conformidades
Autenticar usuario	Iteración 1	El usuario no fue registrado en la base de datos.
	Iteración 2	Desarrollada satisfactoriamente.
Registrar usuario	Iteración 1	El usuario se registró en el sistema con números en el campo nombre y la aplicación no fue capaz de enviar un mensaje de alerta.
	Iteración 2	Desarrollada satisfactoriamente
Crear documento	Iteración 1	El usuario subió un archivo .pdf y el sistema no mostró un mensaje de error.
	Iteración 2	El usuario dio clic en el botón "Enviar" para confirmar el envío, el archivo se guardó con éxito y el campo nombre estaba en blanco, y el sistema no emitió un mensaje de error.
	Iteración 3	Desarrollada satisfactoriamente
Exportar resultados	Iteración 1	Se exportó el documento con extensión .txt y la aplicación no detectó la falla.
	Iteración 2	Desarrollada satisfactoriamente

Tabla 8. Resultados de las pruebas

Se realizaron pruebas de carga para 25 usuarios conectados concurrentemente en un período de subida de un segundo manipulado un documento por cada uno de ellos, demostrando que la aplicación es capaz de soportarlos.

Las pruebas fueron realizadas en una PC con requerimientos de 4GB de RAM y un procesador Intel COREi3 de tercera generación. En la figura 18, se muestra el resultado de las pruebas aplicadas con la herramienta JMeter:

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
/documentos	25	95	16	513	12	514	0,00%	17,3/sec	40,9
/documento-edi...	25	754	756	1274	81	1497	0,00%	9,2/sec	63,3
/documento-mo...	25	833	811	1204	256	1453	0,00%	7,9/sec	54,6
/categoria-most...	25	637	656	911	166	1175	0,00%	8,1/sec	56,1
Total	100	580	626	982	12	1497	0,00%	29,2/sec	168,2

Figura 18. Reporte de la prueba de carga

De acuerdo al resultado arrojado en el reporte de la prueba de carga se puede decir, que para 25 usuarios conectados concurrentemente en un período de subida de un segundo manipulado, un documento por cada uno de ellos, queda demostrado que el rendimiento medio es de 30 segundos.

4.7 Conclusiones parciales

En el presente capítulo se abordó sobre la implementación de la solución para definir la descripción de los elementos físicos que integran el sistema y la topología hardware sobre la que se ejecuta el sitio a través del diagrama de componentes y el diagrama de despliegue respectivamente. Se especificaron los estándares de codificación para guiar el desarrollo y el uso de buenas prácticas de diseño y programación web. Los resultados de las pruebas de software demuestran que se logró garantizar la calidad de la aplicación web y la revisión final del cumplimiento de los requisitos.

CONCLUSIONES GENERALES

Como resultado de la fundamentación teórica, los elementos de la propuesta de solución, el diseño, la implementación y validación del sistema es posible llegar a las siguientes conclusiones:

- En el estudio realizado se detectó que en la Universidad no existe una herramienta para la recopilación y explotación⁴⁰ de los comentarios de los clientes que utilizan las aplicaciones de software desarrolladas por la UCI⁴¹.
- El diagnóstico de la implementación de la minería de texto y la minería de opinión fue el elemento base para poder comprender los aspectos que maneja el aprendizaje automático en la clasificación de los comentarios de usuarios.
- Se determinó la necesidad de crear una aplicación web centrada en el análisis de opiniones sobre aplicaciones de software con el objetivo de utilizar las revisiones de los usuarios que explotan las funcionalidades del sistema, porque las soluciones existentes no cumplen con los requerimientos identificados.
- Se desarrolló la aplicación web AOpinion, que permite que el usuario suba un archivo con extensión .txt que contenga los comentarios de usuarios para ser clasificados en positivo, negativo y neutro, además de mostrar por categorías los comentarios positivos y negativos.
- Fueron realizadas una serie de pruebas, las cuales arrojaron resultados satisfactorios, pues se identificaron un grupo de no conformidades que fueron corregidas, lo que posibilitó la verificación y validación de las funcionalidades del sistema.

⁴⁰ Conjunto de elementos o instalaciones destinados a sacar provecho de un producto natural.

⁴¹ Universidad de las Ciencias Informáticas.

RECOMENDACIONES

- Incluir en la aplicación AOpinion la posibilidad de buscar y recopilar información sobre una aplicación específica en redes sociales como Twitter o Facebook.
- Agregarle a la aplicación AOpinion la posibilidad de procesar los comentarios de usuarios desde archivos en diferentes formatos (por ejemplo, archivos xml o csv).
- Crear un portal o incluir funcionalidades en la aplicación AOpinion para que los usuarios puedan escribir sus experiencias y criterios sobre un software determinado, ejemplo de ello el sistema operativo Nova.

REFERENCIAS

2016. ABIZTAR. [En línea] 2016. [Citado el: 02 de 02 de 2016.]

<http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.

Allic, Mickel. 2010. The Importance of User Feedback. *PixelTango*. [En línea] 2010. [Citado el: 04 de 04 de

2016.] <https://www.pixeltango.com/articles/interaction-design/the-importance-of-user-feedback/>.

Álvarez, Miguel Ángel. 2009. Desarrolloweb.com. *Desarrolloweb.com*. [En línea] 14 de 10 de 2009. [Citado el:

03 de 04 de 2016.] <http://www.desarrolloweb.com/articulos/que-es-html5.html>.

Anton, César. 2015. Platzi. *Platzi*. [En línea] 09 de 07 de 2015. [Citado el: 06 de 04 de 2016.]

<https://platzi.com/blog/que-es-postgresql/>.

Aplicaciones del procesamiento del lenguaje natural en la recuperación de información en español. Vilares

Ferro, Jesús. 2005. 36, A Curuña : s.n., 2005. ISSN 1135-5948.

Aristegui O, José Luis. 2010. Test cases in software test. 2010.

Benavides Cañón, Paula Andrea y Rodríguez Correo, Sandra. *Procesamiento del lenguaje natural en la*

recuperación de información. Bogotá : s.n.

Binali, Haji, Potdar, Vidyasagar y Wu, Chen. 2009. *A State Of The Art Opinion Mining And Its Application*

Domains. Australia : s.n., 2009.

Booch, Grady, Rumbaugh, Jim y Jacobson, Ivar. *El Lenguaje Unificado de Modelado*.

Cortizo Pérez, José Carlos. 2015. BrainSINS. *BrainSINS*. [En línea] 2015. [Citado el: 30 de 01 de 2016.]

<http://www.brainsins.com/es/blog/mineria-opiniones/3555>.

Cridler, Michael. 2014. Narcissistic Jerks Are Giving Play Store Apps 1-Star Reviews For Higher Visibility. *Android*

Police. [En línea] 12 de 01 de 2014. [Citado el: 18 de 01 de 2016.]

<http://www.androidpolice.com/2014/01/12/narcissistic-jerks-are-giving-play-store-apps-1-star-reviews-for-higher-visibility/>.

Damián. 2010. CSS3 y HTML5. *CSS3 y HTML5*. [En línea] 23 de 11 de 2010. [Citado el: 5 de 4 de 2016.]

<http://html5.dwebapps.com/que-es-css3/>.

Desarrolloweb.com. *Desarrolloweb.com*. [En línea] [Citado el: 05 de 04 de 2016.]

<http://www.desarrolloweb.com/manuales/manual-jquery.html>.

Di Meglio, José . 2015. ProgramaciónExtrema.com. *ProgramaciónExtrema.com*. [En línea] 15 de 10 de 2015.

[Citado el: 04 de 04 de 2016.] <http://programacionextrema.com/2015/10/15/crear-un-pdf-en-java-script-jspdf/>.

2016. Django en español. *Django en español*. [En línea] 2016. [Citado el: 01 de 03 de 2016.] <http://django.es>.

2013. enbolivia.com. *enbolivia.com*. [En línea] 01 de 03 de 2013. [Citado el: 05 de 04 de 2016.]
<https://enboliviacom.wordpress.com/2013/03/01/highcharts-libreria-para-creacion-de-graficos/>.

Fayyad, Usama, Piatetsky-Shapiro, Gregory y Smyth, Paradhraic. 1996. *From Data Mining to Knowledge Discovery in Databases*. 1996.

Feldman, Ronen y Dagan, Ido. 1995. *Knowledge Discovery in Textual Databases*. Ramat Gan : s.n., 1995.

Feldman, Ronen y Sanger, James. 2007. *The Text Mining Handbook*. Cambridge : s.n., 2007.

Fontela, Alvaro. Raiola Networks. *Raiola Networks*. [En línea] [Citado el: 05 de 05 de 2016.]
<https://raiolanetworks.es/blog/que-es-bootstrap/>.

Gauchat, Juan Diego. 2012. *El gran libro de HTML5, CSS3 y Javascript*. Barcelona : s.n., 2012. ISBN eBook: 978-84-267-1782-5.

González Duque, Raúl. 2015. *Python para todos*. 2015.

Hernández Orallo, José, Ferri Ramírez, César y Ramírez Quintana, M. José. 2004. *Introducción a la Minería de Datos*. Madrid : Pearson Educación, 2004. 28042.

Hernández, Francisco Refugio Zavala. 2014. *Buscador de artículos científicos aplicando Minería de datos*. México : s.n., 2014.

Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet. **Ordoñez Leyva, Yoanni y Avilés Vázquez, Ernesto. 2010.** La Habana : s.n., 2010.

IBM. 2012. *Manual CRISP-DM de IBM SPSS Modeler*. 2012.

Infante Montero, Sergio. 2012. *Curso Django*. 2012. CC BY-NC-SA 3.0.

2013. iProspect. *iProspect*. [En línea] 06 de 11 de 2013. [Citado el: 20 de 03 de 2016.]
<http://www.iprospect.com/en/ca/blog/10-sentiment-analysis-tools-track-social-marketing-success/>.

JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. 2000. *El Proceso Unificado De Desarrollo de Software*. Madrid: Addison Wesley : s.n., 2000.

Jojo. 2011. How important is feedback? *UX User Experience StackExchange*. [En línea] 30 de 07 de 2011. [Citado el: 19 de 01 de 2016.] <http://ux.stackexchange.com/questions/9472/how-important-is-feedback>.

Larman, Craig. 1999. *UML y patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN 970-17-0261-1.

Liu, Bing. 2010. *Sentiment Analysis and Subjectivity*. s.l. : N.Indurkha and F.J.Damerou, 2010.

Martínez, Rafael. 2013. PostgreSQL-es. *PostgreSQL-es*. [En línea] 2013. [Citado el: 19 de 01 de 2016.]
http://www.postgresql.org.es/sobre_postgresql.

Mendoza, Milagros. 2014. Tienda Nube. *Tienda Nube*. [En línea] 23 de 06 de 2014. [Citado el: 15 de 04 de 2016.] <https://www.tiendanube.com/blog/herramientas-y-consejos-de-monitoreo-redes-sociales/>.

METODOLOGIAS AGILES PROCESO UNIFICADO AGIL (AUP). **FLORES, ERVIN y CORDERO L., JORGE LUIS.** LA PAZ, EL ALTO – BOLIVIA : s.n.

Metodologías Ágiles. Proceso Unificado Ágil (AUP). **Flores, Ervin y Cordero L., Jorge Luis.** La Paz : s.n.

Minería de Datos como soporte a la toma de decisiones empresarial. **Marcano Aular, Yelitza Josefina y Talavera Pereira, Rosalba. 2007.** 52, 2007.

Moine, Juan Miguel. 2013. Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo. 2013.

Object Management Group. *Object Management Group*. [En línea] [Citado el: 06 de 05 de 2016.] <http://www.bpmn.org/>.

Opinion Mining and SentimentAnalysis. **Pang, Bo y Lee, Lilian. 2008.** 1, 2008, Vol. 2.

Pérez Valdés, Damián. 2007. MAESTROS DEL WEB. *MAESTROS DEL WEB*. [En línea] 03 de 07 de 2007. [Citado el: 03 de 04 de 2016.] <http://www.maestrosdelweb.com/que-es-javascript/>.

PosNeg opinion: Una herramienta para gestionar comentarios de la web. **Amores, Mario, Arco, Leticia y Artilles, Michel. 2015.** 1, Villa Clara : s.n., 2015, Vol. 9.

Pressman, Roger S. 2000. *Ingeniería de software. Un enfoque práctico*. Madrid : s.n., 2000. ISBN:84-481-1186-9.

—. **2006.** *Ingeniería del Software. Un enfoque Práctico*. Madrid: McGraw-Hill : Sexta Edición, 2006. ISBN: 970-10-5473-3.

—. **2005.** *Ingeniería del software. Un enfoque práctico. Sexta edición*. 2005.

Pressman, Roger S. 2010. *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. Séptima edición*. Impreso en México : s.n., 2010. ISBN: 978-607-15-0314-5.

—. **2005.** *Ingeniería del Software. Un enfoque Práctico*. 2005. ISBN 970-10-5473-3.

Proceso de pruebas para productos de software en un laboratorio de calidad. **Jústiz-Núñez, Dalila, Gómez-Suárez, Darlene y Delgado-Dapena, Marta Dunia. 2014.** 02, La Habana : s.n., 2014, Ingeniería Industrial, Vol. 35. 1815-5936.

Roldan Diego. 2009. Diagrama de Componentes. *Diagrama de Componentes*. [En línea] 26 de 02 de 2009. [Citado el: 30 de 04 de 2016.] <http://exposicionuml40099.blogspot.com/>.

Sentiment Analysis and Opinion Mining: A Survey. **Vinodhini, G y Chandrasekaran, RM. 2012.** 6, India : s.n., 2012, Vol. 2.

Sistema para la integración de datos bibliográficos almacenados en diversos formatos. **Anel , Pérez Acosta. 2013.** La Habana : s.n., 2013.

Vallejos, Sofía J. 2006. *Diseño y Administración de Datos.* Argentina : s.n., 2006.

BIBLIOGRAFÍA

2016. ABIZTAR. [En línea] 2016. [Citado el: 02 de 02 de 2016.]

<http://www.milestone.com.mx/CursoModeladoNegociosBPMN.htm>.

Allic, Mickel. 2010. The Importance of User Feedback. *PixelTango*. [En línea] 2010. [Citado el: 04 de 04 de 2016.] <https://www.pixeltango.com/articles/interaction-design/the-importance-of-user-feedback/>.

Álvarez, Miguel Ángel. 2009. Desarrolloweb.com. *Desarrolloweb.com*. [En línea] 14 de 10 de 2009. [Citado el: 03 de 04 de 2016.] <http://www.desarrolloweb.com/articulos/que-es-html5.html>.

Anton, César. 2015. Platzi. *Platzi*. [En línea] 09 de 07 de 2015. [Citado el: 06 de 04 de 2016.]

<https://platzi.com/blog/que-es-postgresql/>.

Aplicaciones del procesamiento del lenguaje natural en la recuperación de información en español. **Vilares Ferro, Jesús. 2005.** 36, A Curuña : s.n., 2005. ISSN 1135-5948.

Aristegui O, José Luis. 2010. Test cases in software test. 2010.

Benavides Cañón, Paula Andrea y Rodríguez Correo, Sandra. *Procesamiento del lenguaje natural en la recuperación de información*. Bogotá : s.n.

Binali, Haji, Potdar, Vidyasagar y Wu, Chen. 2009. *A State Of The Art Opinion Mining And Its Application Domains*. Australia : s.n., 2009.

Booch, Grady, Rumbaugh, Jim y Jacobson, Ivar. *El Lenguaje Unificado de Modelado*.

Cortizo Pérez, José Carlos. 2015. BrainSINS. *BrainSINS*. [En línea] 2015. [Citado el: 30 de 01 de 2016.]

<http://www.brainsins.com/es/blog/mineria-opiniones/3555>.

Crider, Michael. 2014. Narcissistic Jerks Are Giving Play Store Apps 1-Star Reviews For Higher Visibility. *Android Police*. [En línea] 12 de 01 de 2014. [Citado el: 18 de 01 de 2016.]

<http://www.androidpolice.com/2014/01/12/narcissistic-jerks-are-giving-play-store-apps-1-star-reviews-for-higher-visibility/>.

Damián. 2010. CSS3 y HTML5. *CSS3 y HTML5*. [En línea] 23 de 11 de 2010. [Citado el: 5 de 4 de 2016.]

<http://html5.dwebapps.com/que-es-css3/>.

Desarrolloweb.com. *Desarrolloweb.com*. [En línea] [Citado el: 05 de 04 de 2016.]

<http://www.desarrolloweb.com/manuales/manual-jquery.html>.

Di Meglio, José . 2015. ProgramaciónExtrema.com. *ProgramaciónExtrema.com*. [En línea] 15 de 10 de 2015. [Citado el: 04 de 04 de 2016.] <http://programacionextrema.com/2015/10/15/crear-un-pdf-en-java-script-jspdf/>.

2016. Django en español. *Django en español*. [En línea] 2016. [Citado el: 01 de 03 de 2016.] <http://django.es>.

2013. enbolivia.com. *enbolivia.com*. [En línea] 01 de 03 de 2013. [Citado el: 05 de 04 de 2016.]
<https://enboliviacom.wordpress.com/2013/03/01/highcharts-libreria-para-creacion-de-graficos/>.

Fayyad, Usama, Piatetsky-Shapiro, Gregory y Smyth, Paradhraic. 1996. *From Data Mining to Knowledge Discovery in Databases*. 1996.

Feldman, Ronen y Dagan, Ido. 1995. *Knowledge Discovery in Textual Databases*. Ramat Gan : s.n., 1995.

Feldman, Ronen y Sanger, James. 2007. *The Text Mining Handbook*. Cambridge : s.n., 2007.

Fontela, Alvaro. Raiola Networks. *Raiola Networks*. [En línea] [Citado el: 05 de 05 de 2016.]
<https://raiolanetworks.es/blog/que-es-bootstrap/>.

Gauchat, Juan Diego. 2012. *El gran libro de HTML5, CSS3 y Javascript*. Barcelona : s.n., 2012. ISBN eBook: 978-84-267-1782-5.

González Duque, Raúl. 2015. *Python para todos*. 2015.

Hernández Orallo, José, Ferri Ramírez, César y Ramírez Quintana, M. José. 2004. *Introducción a la Minería de Datos*. Madrid : Pearson Educación, 2004. 28042.

Hernández, Francisco Refugio Zavala. 2014. *Buscador de artículos científicos aplicando Minería de datos*. México : s.n., 2014.

Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet. **Ordoñez Leyva, Yoanni y Avilés Vázquez, Ernesto. 2010.** La Habana : s.n., 2010.

IBM. 2012. *Manual CRISP-DM de IBM SPSS Modeler*. 2012.

Infante Montero, Sergio. 2012. *Curso Django*. 2012. CC BY-NC-SA 3.0.

2013. iProspect. *iProspect*. [En línea] 06 de 11 de 2013. [Citado el: 20 de 03 de 2016.]
<http://www.iprospect.com/en/ca/blog/10-sentiment-analysis-tools-track-social-marketing-success/>.

JACOBSON, Ivar, BOOCH, Grady y RUMBAUGH, James. 2000. *El Proceso Unificado De Desarrollo de Software*. Madrid: Addison Wesley : s.n., 2000.

Jojo. 2011. How important is feedback? *UX User Experience StackExchange*. [En línea] 30 de 07 de 2011. [Citado el: 19 de 01 de 2016.] <http://ux.stackexchange.com/questions/9472/how-important-is-feedback>.

Larman, Craig. 1999. *UML y patrones. Introducción al análisis y diseño orientado a objetos*. México : Prentice Hall, 1999. ISBN 970-17-0261-1.

Liu, Bing. 2010. *Sentiment Analysis and Subjectivity*. s.l. : N.Indurkha and F.J.Damerou, 2010.

Martínez, Rafael. 2013. PostgreSQL-es. *PostgreSQL-es*. [En línea] 2013. [Citado el: 19 de 01 de 2016.]
http://www.postgresql.org.es/sobre_postgresql.

Mendoza, Milagros. 2014. Tienda Nube. *Tienda Nube*. [En línea] 23 de 06 de 2014. [Citado el: 15 de 04 de 2016.] <https://www.tiendanube.com/blog/herramientas-y-consejos-de-monitoreo-redes-sociales/>.

METODOLOGIAS AGILES PROCESO UNIFICADO AGIL (AUP). **FLORES, ERVIN y CORDERO L., JORGE LUIS.** LA PAZ, EL ALTO – BOLIVIA : s.n.

Metodologías Ágiles. Proceso Unificado Ágil (AUP). **Flores, Ervin y Cordero L., Jorge Luis.** La Paz : s.n.

Minería de Datos como soporte a la toma de decisiones empresarial. **Marcano Aular, Yelitza Josefina y Talavera Pereira, Rosalba. 2007.** 52, 2007.

Moine, Juan Miguel. 2013. Metodologías para el descubrimiento de conocimiento en bases de datos: un estudio comparativo. 2013.

Object Management Group. *Object Management Group*. [En línea] [Citado el: 06 de 05 de 2016.] <http://www.bpmn.org/>.

Opinion Mining and SentimentAnalysis. **Pang, Bo y Lee, Lilian. 2008.** 1, 2008, Vol. 2.

Pérez Valdés, Damián. 2007. MAESTROS DEL WEB. *MAESTROS DEL WEB*. [En línea] 03 de 07 de 2007. [Citado el: 03 de 04 de 2016.] <http://www.maestrosdelweb.com/que-es-javascript/>.

PosNeg opinion: Una herramienta para gestionar comentarios de la web. **Amores, Mario, Arco, Leticia y Artilles, Michel. 2015.** 1, Villa Clara : s.n., 2015, Vol. 9.

Pressman, Roger S. 2000. *Ingeniería de software. Un enfoque práctico*. Madrid : s.n., 2000. ISBN:84-481-1186-9.

—. **2006.** *Ingeniería del Software. Un enfoque Práctico*. Madrid: McGraw-Hill : Sexta Edición, 2006. ISBN: 970-10-5473-3.

—. **2005.** *Ingeniería del software. Un enfoque práctico. Sexta edición*. 2005.

Pressman, Roger S. 2010. *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO. Séptima edición*. Impreso en México : s.n., 2010. ISBN: 978-607-15-0314-5.

—. **2005.** *Ingeniería del Software. Un enfoque Práctico*. 2005. ISBN 970-10-5473-3.

Proceso de pruebas para productos de software en un laboratorio de calidad. **Jústiz-Núñez, Dalila, Gómez-Suárez, Darlene y Delgado-Dapena, Marta Dunia. 2014.** 02, La Habana : s.n., 2014, Ingeniería Industrial, Vol. 35. 1815-5936.

Roldan Diego. 2009. Diagrama de Componentes. *Diagrama de Componentes*. [En línea] 26 de 02 de 2009. [Citado el: 30 de 04 de 2016.] <http://exposicionuml40099.blogspot.com/>.

Sentiment Analysis and Opinion Mining: A Survey. **Vinodhini, G y Chandrasekaran, RM. 2012.** 6, India : s.n., 2012, Vol. 2.

Sistema para la integración de datos bibliográficos almacenados en diversos formatos. **Anel , Pérez Acosta. 2013.** La Habana : s.n., 2013.

Vallejos, Sofía J. 2006. *Diseño y Administración de Datos.* Argentina : s.n., 2006.

ANEXOS

Anexo #1. Entrevista realizada al Director del Centro de Soporte de la UCI

Esta entrevista se realizó con el objetivo de entender el proceso de revisiones de los softwares⁴² en el Centro de Soporte. Ganar en claridad en cuanto a cómo se gestionan los comentarios de los clientes para conocer los errores encontrados, algunas recomendaciones y sus criterios durante el uso de la aplicación.

1. ¿A qué se dedica el Centro de Soporte?
2. ¿A qué servicios le brinda soporte dicho Centro?
3. ¿Se tienen en cuenta la opinión de los clientes que utilizan las aplicaciones? ¿Sí o no, y por qué?

El resultado de la entrevista ayudó a fundamentar teóricamente la investigación. Se pudo apreciar que en el Centro de Soporte se usa la metodología ITIL, por lo que no se tiene en cuenta la opinión de los clientes que hacen uso de los diferentes sistemas, ya que solo se generan las incidencias y problemas pertinentes al producto.

Anexo #2: Especificación de los requisitos funcionales.

En la tabla 9, se muestra la especificación del RF#2 “Registrar usuario”.

Precondiciones	1. El usuario no ha sido registrado en el sistema.
Flujo de eventos	
Flujo básico	
1. Entrar a la página de inicio.	
2. Presionar el botón “Registrarse”.	
3. Insertar los datos de los campos: Nombre, Usuario, Email, Contraseña y Repetir contraseña.	
4. Confirmar los datos escritos haciendo clic en el botón “Enviar”.	
Pos-condiciones	
1. El usuario queda registrado en el sistema.	
Validaciones	

⁴² Equipo lógico o soporte lógico de un sistema informático.

Validaciones
1. El usuario debe escribir su nombre comenzando por la letra inicial mayúscula.
2. El usuario inserta su contraseña con un mínimo de 6 caracteres.
3. El usuario no debe escribir números en el campo: nombre.
Pos-condiciones
1. Ir a la descripción del RF #1: “Autenticar usuario”.
Prototipo de interfaz


Tabla 9. Especificación del RF#2 “Registrar usuario”

En la tabla 10, se muestra la especificación del RF#3 “Gestionar documento”.

Precondiciones	1. El usuario se autenticó en el sistema.
Flujo de eventos	
Flujo básico	
1. Mostrar el menú superior.	
2. Mostrar las opciones de Agregar Documento, Editar Documento, Mostrar Resultados y Eliminar Documento.	
3. Se muestra una tabla con el nombre del documento, la fecha del día que se subió, la precisión con la que se clasificaron los comentarios y el usuario de la persona.	
Flujos alternativos	
Flujos alternativos 3.a Agregar Documento	

1. En caso de seleccionar la opción Agregar Documento. Ver el RF #3.1 “Crear documento”.

Flujos alternativos 3.b Editar Documento

1. En caso de seleccionar la opción Editar Documento. Ver el RF #3.2 “Editar documento”.

Flujos alternativos 3.c Mostrar Resultados

1. En caso de seleccionar la opción Mostrar Documento. Ver el RF #3.4 “Mostrar resultados”.

Flujos alternativos 3.d Eliminar Documento

1. En caso de seleccionar la opción Eliminar Documento. Ver el RF #3.4 “Eliminar documento”.

Prototipo de interfaz



Tabla 10. Especificación del RF#3 “Gestionar documento”

En la tabla 11, se muestra la especificación del RF#3.1 “Crear documento”.

Precondiciones	<ol style="list-style-type: none"> 1. El usuario se autenticó en el sistema. 2. El usuario desea crear un documento.
Flujo de eventos	
Flujo básico	
1. Presionar el botón “Agregar Documento”.	
2. Llenar el campo: Nombre.	
3. Presionar el botón “Examinar “.	
4. Escoger el archivo a subir.	

5. Presionar el botón “Abrir”.

6. Confirmar el envío del archivo haciendo clic en el botón “Enviar”.

Validaciones

1. El usuario debe escoger un archivo que tenga extensión .txt.

Prototipo de interfaz

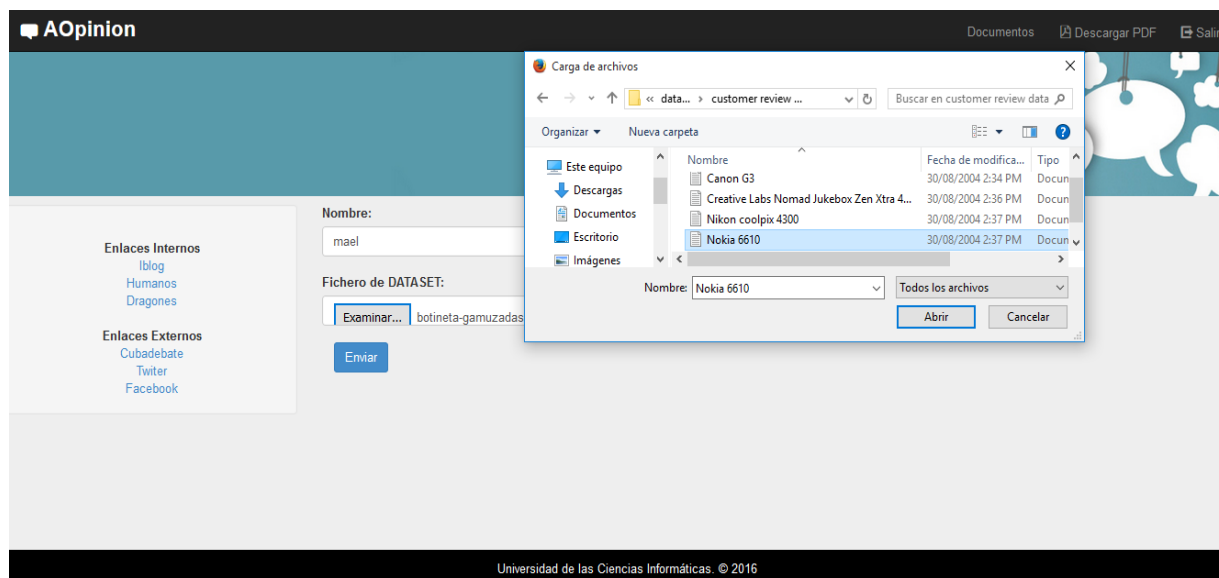


Tabla 11. Especificación del RF#3.1 “Crear documento”

En la tabla 12, se muestra la especificación del RF#3.2 “Editar documento”.

Precondiciones	1. El usuario se autenticó en el sistema.
Flujo de eventos	
Flujo básico	
1. Presionar el botón “Editar Documento”.	
2. Llenar el campo: Nombre.	
3. Presionar el botón “Examinar”.	
4. Escoger el archivo a subir.	
5. Presionar el botón “Abrir”.	
6. Confirmar el envío del archivo haciendo clic en el botón “Enviar”.	
Validaciones	
1. Solo se permite escoger un fichero .txt.	



Tabla 12. Especificación del RF#3.2 “Editar documento”

En la tabla 13, se muestra la especificación del RF#3.3 “Eliminar documento”.

Precondiciones	<ol style="list-style-type: none"> 1. El usuario se autenticó en el sistema. 2. Se tiene al menos un documento en el registro de documentos.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. Seleccionar un documento. 2. Presionar el botón “Eliminar Documento” 	

Tabla 13. Especificación del RF#3.3 “Eliminar documento”

En la tabla 14, se muestra la especificación del RF#5 “Mostrar resultados”.

Precondiciones	<ol style="list-style-type: none"> 1. El usuario se autenticó en el sistema. 2. El usuario desea ver un documento de los que tiene en el sistema.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. Seleccionar un documento. 	

2. Presionar el botón “Mostrar Resultados”.
3. Se muestra un menú superior con las opciones de Gráfica Categorías y Gráfica Comentarios y debajo una tabla con el nombre del documento, la fecha del día que se subió, la precisión con la que se clasificaron los comentarios y el usuario de la persona. Además se muestra un menú inferior.
Flujos alternativos
Flujos alternativos 3.a Gráfica Categorías
1. En caso de seleccionar en el menú superior “Gráfica Categorías”: Ver el RF #5.1 “Gráfica categorías”.
Flujos alternativos 3.b Gráfica Comentarios
1. En caso de seleccionar en el menú superior “Gráfica Comentarios”: Ver el RF #5.2 “Gráfica comentarios”.
Flujos alternativos 3.c Categorías
1. En el menú inferior se muestra una tabla con el número, nombre de las categorías que tienen los comentarios, la cantidad de comentarios positivos y negativos que tiene el documento. Además, se muestran las acciones, donde por cada categoría que se genera un botón “Mostrar”.
2. En caso de seleccionar “Mostrar”: Ver el RF #5.3 “Listar categorías”.
Flujos alternativos 3.d Comentarios Positivos
1. En caso de seleccionar la opción “Comentarios Positivos” en el menú inferior se muestran todos los comentarios positivos con la polaridad, para conocer qué tan positivo y negativo es el comentario, la subjetividad (para saber qué tan objetivo ⁴³ y subjetivo es el sentimiento) y la clasificación que devuelve el clasificador.
Flujos alternativos 3.e Comentarios Negativos
1. En caso de seleccionar la opción “Comentarios Negativos” en el menú inferior se muestran todos los comentarios negativos con la polaridad, para conocer qué tan positivo y negativo es el comentario, la subjetividad (para saber qué tan objetivo y subjetivo es el sentimiento) y la clasificación que resulta del clasificador.
Flujos alternativos 3.f Comentarios Neutros
1. En caso de seleccionar la opción “Comentarios Neutros” en el menú inferior se muestran todos los comentarios neutros con la polaridad, para conocer qué tan positivo y negativo es

⁴³ Es lo relativo al objeto en sí, independientemente de juicios personales, lo que no es subjetivo, y que no se deja influir por consideraciones personales en sus juicios o en su comportamiento.

el comentario, la subjetividad (para saber qué tan objetivo y subjetivo es el sentimiento) y la clasificación que resulta del clasificador.

Prototipo de interfaz



Tabla 14. Especificación del RF#5 “Mostrar resultados”.

En la tabla 15, se muestra la especificación del RF#5.1 “Graficar categorías”.

Precondiciones	<ol style="list-style-type: none"> 1. El usuario se autenticó en el sistema. 2. Se tiene al menos un documento en el registro de documentos. 3. El usuario presionó el botón “Mostrar Resultados”.
Flujo de eventos	
Flujo básico	
1. Se muestra una gráfica donde se encuentra la cantidad de comentarios positivos y negativos por categorías.	
Flujos alternativos	
Flujos alternativos 3.a Positivos	
1. En caso de seleccionar el botón “Positivos” se muestran en la gráfica solo la cantidad de comentarios positivos por categorías.	
Flujos alternativos	
Flujos alternativos 3.a Negativos	

1. En caso de seleccionar el botón “Negativos” se muestran en la gráfica solo la cantidad de comentarios Negativos por categorías.

Prototipo de interfaz

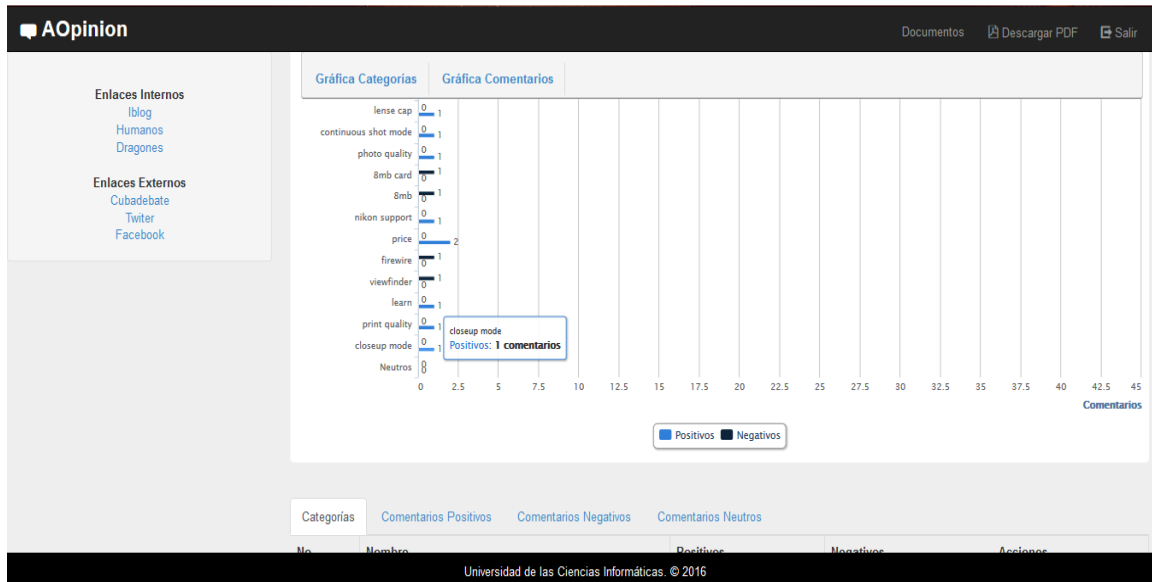


Tabla 15. Especificación del RF#5.1 “Graficar categorías”.

En la tabla 16, se muestra la especificación del RF#5.2 “Graficar comentarios”.

Precondiciones	<ol style="list-style-type: none"> 1. El usuario se autenticó en el sistema. 2. Se tiene al menos un documento en el registro de documentos. 3. El usuario presionó el botón “Mostrar Resultados”.
Pos-condiciones	
<ol style="list-style-type: none"> 1. Se muestra una gráfica donde se encuentra la cantidad de comentarios positivos, negativos y neutros. 	
Prototipo de interfaz	



Tabla 16. Especificación del RF #5.2” Graficar comentarios”

En la tabla 17, se muestra la especificación del RF#5.3 “**Listar categorías**”.

Precondiciones	<ol style="list-style-type: none"> 1. El usuario se autenticó en el sistema. 2. Se tiene al menos un documento en el registro de documentos. 3. El usuario desea ver la polaridad de los comentarios que tiene un documento almacenado en el sistema.
Flujo de eventos	
Flujo básico	
1. Se muestran por categorías todos los comentarios positivos, negativos y neutros.	
2. Se muestra un menú con las opciones “Comentarios Positivos”, “Comentarios Negativos” y “Comentarios Neutros”.	
Flujos alternativos	
Flujos alternativos 2.a Comentarios Positivos	
1. En caso de seleccionar la opción “Comentarios Positivos” en el menú inferior se muestran todos los comentarios positivos con la polaridad, para conocer qué tan positivo, negativo y neutro es el comentario, la subjetividad (para saber qué tan objetivo y subjetivo es el sentimiento) y la clasificación que resulta del clasificador.	
Flujos alternativos 2.b Comentarios Negativos	

1. En caso de seleccionar la opción “Comentarios Negativos” en el menú inferior se muestran todos los comentarios negativos con la polaridad, para conocer qué tan positivo, negativo y neutro es el comentario, la subjetividad (para saber qué tan objetivo y subjetivo⁴⁴ es el sentimiento) y la clasificación que resulta del clasificador.

Flujos alternativos 2.c Comentarios Neutros

1. En caso de seleccionar la opción “Comentarios Neutros” en el menú inferior se muestran todos los comentarios neutros con la polaridad, para conocer qué tan positivo, negativo y neutro es el comentario, la subjetividad (para saber qué tan objetivo y subjetivo es el sentimiento) y la clasificación que resulta del clasificador.

Prototipo de interfaz

The screenshot shows the AOpinion web interface. At the top, there is a navigation bar with 'AOpinion', 'Documentos', 'Descargar PDF', and 'Salir'. On the left, there are links for 'Enlaces Internos' (Iblog, Humanos, Dragones) and 'Enlaces Externos' (Cubadebate, Twitter, Facebook). The main content area displays document details: 'Nombre de Documento: lop', 'Fecha: June 3, 2016', 'Precisión: 0.27500', and 'Usuario: admin'. Below this, the category is 'Categoría: software'. There are three tabs: 'Comentarios Positivos', 'Comentarios Negativos', and 'Comentarios Neutros'. The 'Comentarios Neutros' tab is active, showing a table of comments.

No	Comentario	Polaridad	Subjetividad	Clasificación
1	the software that comes with it is amazing and the online service that comes free is really very neat. clean clear and well focused on over 95 % of all photos taken by a beginner .	0.4	0.66666666667	neu
2	the software you get with this camera is perfect .	1.0	1.0	neg
3	the 4300 comes with excellent easy to install software i use it on xp and win98 without problem , battery charger , usb and video cable .	0.71666666667	0.91666666667	neu

Universidad de las Ciencias Informáticas. © 2016

Tabla 17. Especificación del RF #5.3 “Listar categorías”.

En la tabla 18, se muestra la especificación del RF#6 “Exportar resultados”.

Precondiciones	<ol style="list-style-type: none"> 1. El usuario se autenticó en el sistema. 2. Se tiene al menos un documento en el registro de documentos.
Flujo de eventos	
Flujo básico	
<ol style="list-style-type: none"> 1. El usuario accede al menú principal de la aplicación y da clic en el botón “Descargar PDF”. 	

⁴⁴ Es un adjetivo que identifica algo como propio de la manera de pensar o sentir de una persona.

2. Se muestra una ventana con las opciones de guardar o abrir el documento.
3. El usuario debe presionar el botón “Aceptar”.

Pos-condiciones

1. Se exportan los resultados obtenidos con extensión pdf.

Prototipo de interfaz

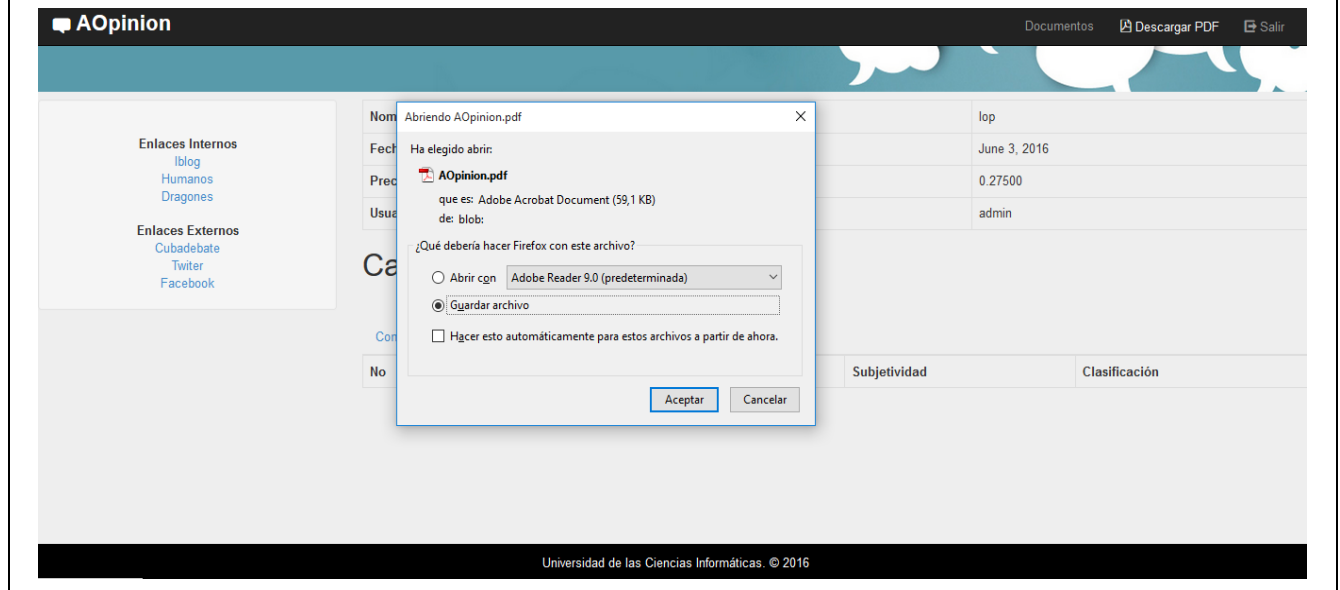


Tabla 18. Especificación del RF #6 “Exportar resultados”.

Anexo #3: Diagramas de clases del diseño

En la figura 19, se muestra el diagrama de clases del diseño del sistema utilizando los estereotipos web, correspondiente al CU: “Registrar usuario”.

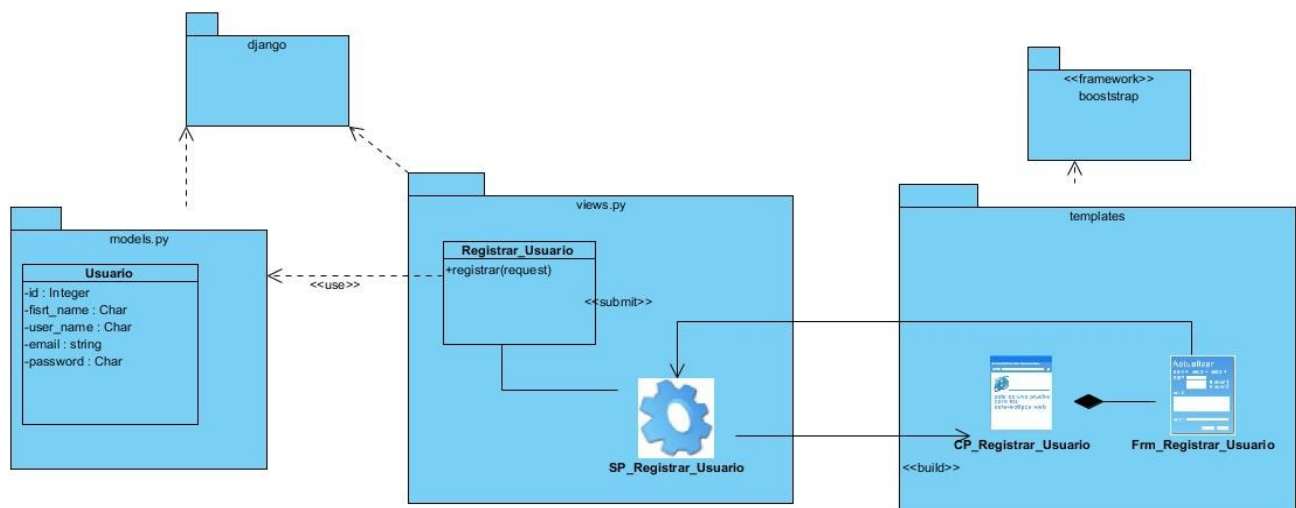


Figura 19. Diagrama de clases del diseño “Registrar usuario”

En la figura 20, se muestra el diagrama de clases del diseño del sistema utilizando los estereotipos web, correspondiente al CU: “Gestionar documento”.

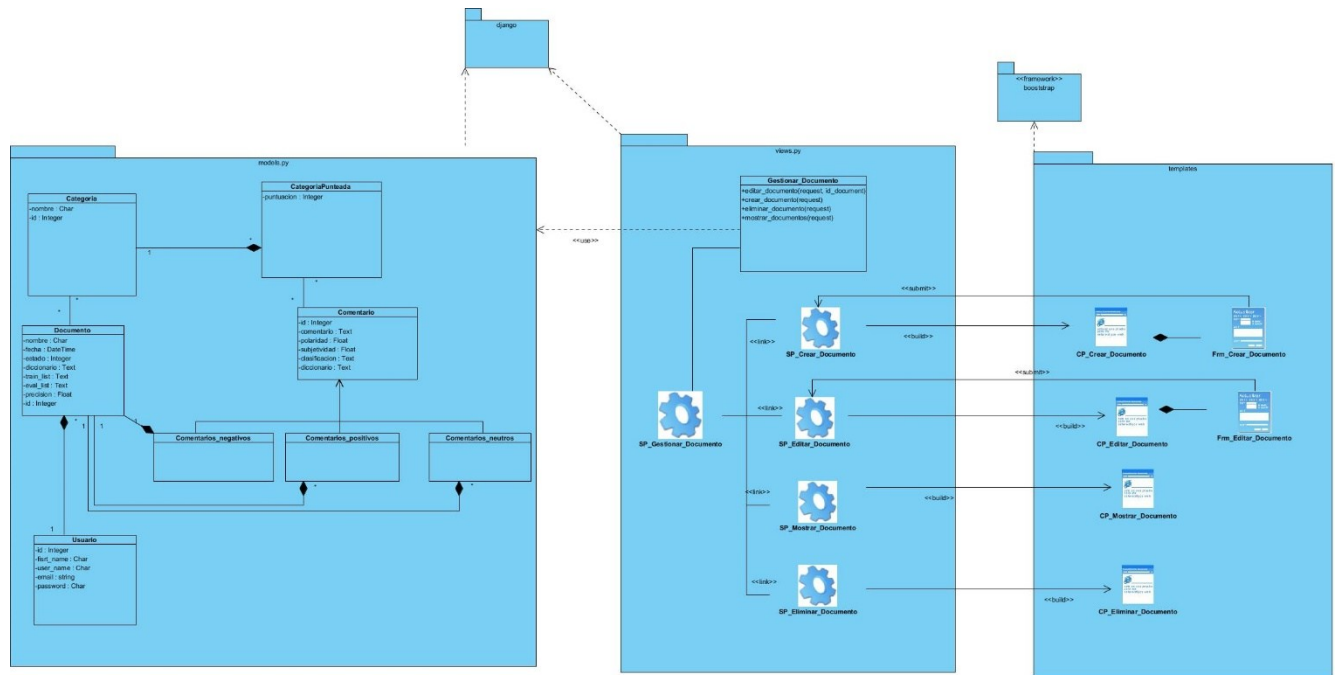


Figura 20. Diagrama de clases del diseño “Gestionar documento”

En la figura 21, se muestra el diagrama de clases del diseño del sistema utilizando los estereotipos web, correspondiente al CU: “Mostrar resultados”.

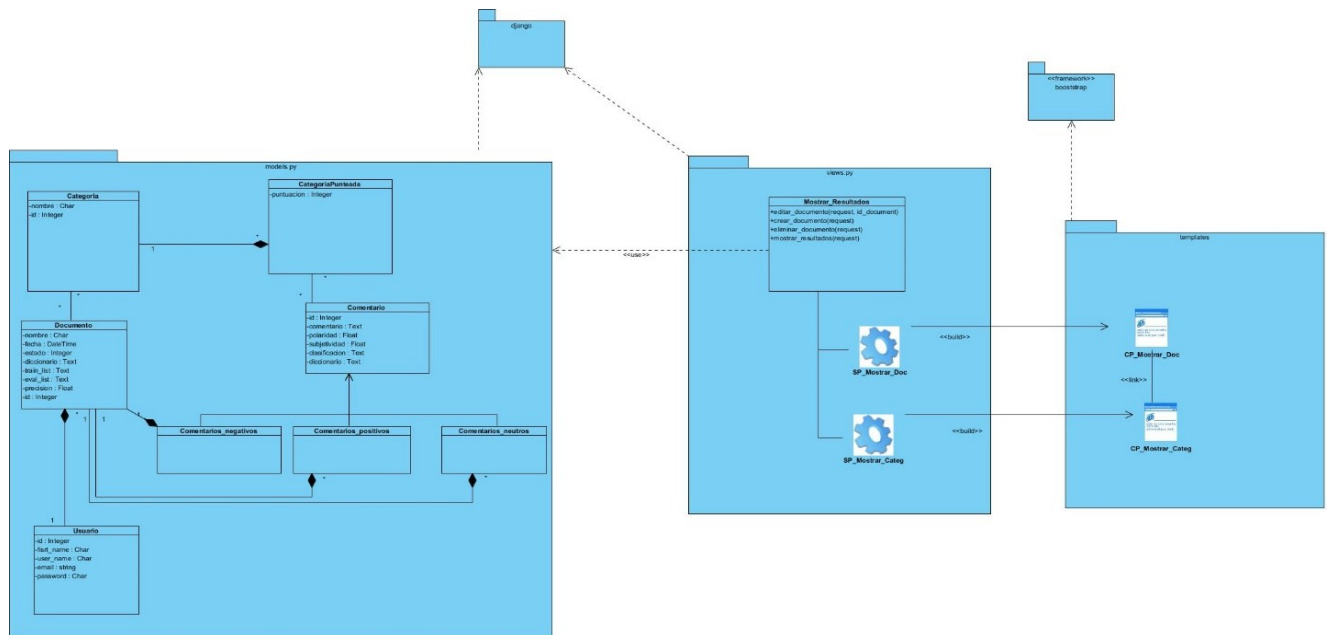


Figura 21. Diagrama de clases del diseño “Mostrar resultados”

Anexo #4 Casos de prueba de los Requisitos funcionales.

En la tabla 19, se muestra el caso de prueba RF#2 “Registrar usuario”.

Caso de prueba RF#2			
Condiciones de ejecución: Registrar usuario			
Claves válidas	Claves no válidas	Resultado esperado	Resultado de prueba
El usuario escribe su nombre(letra inicial mayúscula y solo letras), usuario, email(los requisitos de email), contraseña(como mínimo 6 caracteres) y repite la contraseña(que coincida con la contraseña) .		El usuario queda registrado en la base de datos de forma correcta.	Satisfactoria
	El usuario escribe con minúscula o con números, incluso ambos.	El usuario recibe el mensaje “El nombre se escribe con mayúscula” o	Satisfactoria

		“El nombre no debe contener números”.	
	El usuario escribe incorrectamente el email.	El usuario recibe el mensaje “El email no es válido”.	Satisfactoria
	El usuario escribe la contraseña con menos de seis caracteres.	El usuario recibe el mensaje “La contraseña debe tener al menos seis caracteres”.	Satisfactoria
	El usuario escribe las contraseñas sin que coincidan.	El usuario recibe el mensaje “Las contraseñas no coinciden”.	Satisfactoria
	El usuario llena el campo usuario con una contraseña existente en la base de datos.	El usuario recibe el mensaje “El usuario ya está registrado”.	

Tabla 19. Caso de prueba del RF#2: “Registrar usuario”

En la tabla 20, se muestra el caso de prueba RF#3.1 “**Crear documento**”.

Caso de prueba RF#3.1			
Condiciones de ejecución: Crear documento			
Claves válidas	Claves no válidas	Resultado esperado	Resultado de prueba
El usuario sube un documento en el formato .txt.		El documento subido queda almacenado en la carpeta media/documents.	Satisfactoria
	El usuario sube un documento que no es extensión .txt.	El documento subido no se almacena en la carpeta media y se muestra el mensaje “Solo está permitido los ficheros con la extensión .txt”	Satisfactoria

Tabla 20. Caso de prueba del RF#3.1: “Crear documento”

En la tabla 21, se muestra el caso de prueba RF#3.2 “**Editar documento**”.

Caso de prueba RF#3.2			
Condiciones de ejecución: Editar documento			
Claves válidas	Claves no válidas	Resultado esperado	Resultado de prueba
El usuario sube un documento en el formato .txt.		El documento subido queda almacenado en la carpeta media/documents.	Satisfactoria
	El usuario sube un documento que no es extensión .txt.	El documento subido no se almacena en la carpeta media y se muestra el mensaje “Solo está permitido los ficheros con la extensión .txt”	Satisfactoria

Tabla 21. Caso de prueba del RF#3.2: “Editar documento”