UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

Presentado por

MABEL VEGA QUISPE

SISTEMA WEB DE RECOMENDACIÓN DE NOTICIAS BASADO EN COTENIDO PARA MEDIOS DE COMUNICACIÓN

Asesor

Ing. RICHARD ARTEMIO FLORES CONDORI

Co - Asesor

Mtro. JUAN JOSÉ ORÉ CERRÓN

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

ANDAHUAYLAS – APURIMAC – PERÚ

2020

APROBACION DEL ASESOR

Quién suscribe:
Ing. RICHARD ARTEMIO FLORES CONDORI:

CERTIFICA,
Que, el Bachiller en Ingeniería de Sistemas, MABEL VEGA QUISPE ha culminado satisfactoriamente el Proyecto de Tesis intitulado: "SISTEMA WEB DE RECOMENDACIÓN DE NOTICIAS BASADO EN CONTENIDO PARA MEDIOS DE COMUNICACION" para optar el Título Profesional de Ingeniero de Sistemas.

Andahuaylas, 16 de diciembre del 2019.

Ing. Richard Artemio Flores Condori
Asesor

Br. Mabel Vega Quispe
Tesista

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS
RESOLUCIÓN N°488-2020-CFI-UNAJMA
RESOLUCIÓN DE COORDINACIÓN DE FACULTAD DE INGENIERÍA

Andahuaylas, 31 de diciembre de 2020

VISTO: El informe N°59-2020-YTPA-P/JET-EPIS-FI- UNAJMA, de fecha 31 de diciembre de 2020, el Dr. Yalmar Ponce Atencio presidente del jurado evaluador, remite el **Acta de Sustentación de Tesis** del Bachiller en Ingeniería de Sistemas **VEGA QUISPE MABEL y;**

CONSIDERANDO:

Que, por Ley Nº 28372 del 29 de octubre del 2004, se crea la Universidad Nacional José María Arguedas, con sede en la provincia de Andahuaylas, Región Apurímac; y que por Resolución Nº 035-2017-SUNEDU/CD de 02 de octubre del 2017, el Consejo Directivo de la Superintendencia Nacional de Educación Superior Universitaria, otorga la Licencia Institucional a la Universidad Nacional José María Arguedas para ofrecer el Servicio Educativo Superior Universitario;

Que, la Ley Universitaria 30220 en su Artículo Octavo respecto a la autonomía universitaria, establece que: "El estado reconoce la autonomía universitaria". La autonomía inherente a las universidades se ejerce de conformidad a la Constitución, las leyes y demás normativa aplicable, esta Normativa se manifiesta en los siguientes regímenes: Normativo, De gobierno, Académico, Administrativo y Económico;

Que, mediante Carta Múltiple Nº 020-2014-SG-UNAJMA, de fecha 30 de julio del 2014; la Secretaría General de la UNAJMA comunica que mediante Acuerdo Nº 03 de Sesión Ordinaria de la Comisión de Gobierno se **AUTORIZA** la emisión de **RESOLUCIONES DE COORDINACIÓN DE LA FACULTAD** estrictamente para asuntos académicos y deberán remitirse un original a la Secretaría General;

Que, mediante carta Nº 236-2016-SG-UNAJMA de fecha 05 de agosto de 2016 el Ing. Enrique Edgardo Cóndor Tinoco, Secretario General de la UNAJMA, comunica que el Dr. Oswaldo Luizar Obregón, presidente de la Comisión Organizadora de la UNAJMA ha dispuesto que las resoluciones emitidas por la Facultad se deriven a la Vicepresidencia Académica;

Que, el **art. 39 incisos a y d del TITULO II, CAPITULO II del Reglamento General de la UNAJMA**, aprobado mediante Resolución Nº 0130-2016-CO-UNAJMA, establece que "Son *funciones de las Facultades: a) dirigir el desarrollo académico y administrativo de las Escuelas Profesionales y Departamentos Académicos adscritos a esta, dentro de la normatividad legal, d) administrar el sistema de matrícula en coordinación y apoyo con la oficina respectiva*";

Que, del CAPÍTULO **VI DE LA TESIS PARA LA OBTENCIÓN DEL TÍTULO PROFESIONAL** del Reglamento General de Grados y Títulos en la Universidad Nacional José María Arguedas, aprobado con Resolución Nº 0207-2020-CO-UNAJMA; en su art. 93º establece que "*La información correspondiente a la sustentación y los resultados de la calificación, serán asentadas en el Acta de Sustentación de Tesis (Formato del Anexo 23: Acta de Sustentación de Tesis), la misma que deberá de ser elevada por el Jurado Evaluador a la Coordinación de la Facultad, para su aprobación mediante acto resolutivo de la Facultad".*

Que, con Resolución Nº277-2020-CFI-UNAJMA, de fecha 02 de setiembre de 2020, se aprueba el Proyecto de Tesis del Bachiller en Ingeniería de Sistemas por la Modalidad de Sustentación de Tesis, el mismo que ha sido **APROBADO SIN OBSERVACIONES** conforme se detalla a continuación

Título de Tesis	"SISTEMA WEB DE RECOMENDACIÓN DE NOTICIAS BASADO EN CONTENIDOS PARA MEDIOS DE COMUNICACIÓN"	
Autor de Tesis	Bachiller en Ingeniería de Sistemas **VEGA QUISPE MABEL**	
Asesor de Tesis	Ing. Richard Artemio Flores Condori	
Co-Asesor	Mtro. Juan José Oré Cerrón	
Jurado Evaluador	Presidente:	Dr. Yalmar T. Ponce Atencio
	Primer Miembro:	Dr. Julio Cesar Huanca Marín
	Segundo Miembro:	Ing. Roberto Quispe Quispe

Que, con Acta de Sustentación de Tesis, de fecha 31 de diciembre del 2020, el Jurado Evaluador presidida por el Dr. Yalmar Temístocles Ponce Atencio, Primer Miembro Dr. Julio Cesar Huanca Marín y Segundo Miembro Ing.

Roberto Quispe Quispe APRUEBAN por UNANIMIDAD con la calificación de DIECISIETE (17) y mención **MUY BUENO,** el Informe Final de Tesis **"SISTEMA WEB DE RECOMENDACIÓN DE NOTICIAS BASADO EN CONTENIDOS PARA MEDIOS DE COMUNICACIÓN"** cuyo autor es el Bachiller en Ingeniería de Sistemas **VEGA QUISPE MABEL**, de acuerdo al artículo 92º del Reglamento General de Grados y Títulos de la Universidad Nacional José María Arguedas aprobado con Resolución N°0207-2020-CO-UNAJMA;

Que, con informe N°59-2020-YTPA-P/JET-EPIS-FI- UNAJMA, de fecha 31 de diciembre de 2020, el Dr. Yalmar Ponce Atencio presidente del jurado evaluador, alcanza a esta coordinación las actas de sustentación de fecha 31 de diciembre del 2020 del Bachiller en Ingeniería de Sistemas **VEGA QUISPE MABEL**.

Que, en atención al Informe N°59-2020-YTPA-P/JET-EPIS-FI- UNAJMA; el Dr. Yalmar Temístocles Ponce Atencio, Coordinador de la Facultad de Ingeniería de la Universidad Nacional José María Arguedas, dispone a la Secretaría Académica de la Facultad de Ingeniería proyectar la Resolución correspondiente, la que se aprueba con cargo a dar cuenta a la Vicepresidencia Académica de la UNAJMA;

Por estos considerandos y en uso de las atribuciones conferidas como Coordinador de la Facultad de Ingeniería, designado mediante Resolución Nº 0298-2019-CO-UNAJMA, de fecha 15 de octubre de 2019;

SE RESUELVE:

ARTÍCULO PRIMERO: APROBAR el Acta de Sustentación de Tesis de fecha 31 de diciembre del 2020 en la que el Jurado Evaluador calificó el Informe Final de Tesis: **"SISTEMA WEB DE RECOMENDACIÓN DE NOTICIAS BASADO EN CONTENIDOS PARA MEDIOS DE COMUNICACIÓN"** cuyo autor es el Bachiller en Ingeniería de Sistemas **VEGA QUISPE MABEL;** el mismo que ha sido **APROBADO POR UNANIMIDAD** con el siguiente calificativo y mención:

NOTA ESCALA VIGESIMAL		MENCIÓN
NÚMEROS	LETRAS	
17	DIECISIETE	MUY BUENO

ARTÍCULO SEGUNDO: ENCARGAR a Escuela Profesional de Ingeniería de Sistemas, ejecute y adopte las acciones académicas y administrativas que correspondan, para el cabal cumplimiento de la presente Resolución.

ARTÍCULO TERCERO: REMITIR la presente Resolución a la Vicepresidencia Académica, Escuela Profesional de Ingeniería Sistemas, Docente Asesor, Miembros de Jurado Evaluador y al interesado para su conocimiento y fines pertinentes.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE.

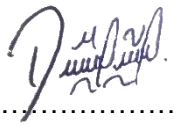
UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS
Dr. Yalmar Ponce Atencio
COORDINADOR DE LA FACULTAD DE INGENIERÍA

Ing. Richard A. Flores Condori
SECRETARIO ACADEMICO

DECLARACIÓN JURADA DE AUTENTICIDAD

Yo, Mabel Vega Quispe, identificado (a) con DNI N°71114848 de la Escuela Profesional de Ingeniería de Sistemas. Declaro bajo juramento que el Proyecto Titulado: "SISTEMA WEB DE RECOMENDACIÓN DE NOTICIAS BASADO EN CONTENIDO PARA MEDIOS DE COMUNICACIÓN" Es auténtico y no vulnera los derechos de autor. Además, su contenido es de entera responsabilidad del autor (es) del proyecto, quedando la UNAJMA exenta de toda responsabilidad en caso de atentar contra la Ley de propiedad intelectual y derechos de autor.

Andahuaylas, 16 de diciembre de 2020



…….
Firma
N° DNI: 71114848
E-mail: mabhexita@gmail.com
N° Celular:926560259

DEDICATORIA

Quiero dedicar el presente informe de tesis, para mi querida madre y mis hermanos en especial para Keyla, quienes con sus palabras de aliento no me dejaron decaer para seguir adelante y siempre sea perseverante y cumpla con mis objetivos.

A mi amado hijo Sayri, que es lo más valioso que la vida me dio y por ser mi fuente de motivación e inspiración. Gracias a ellos y la ayuda de Dios pude lograr que mis sueños se hagan realidad.

AGRADECIMIENTOS

En primera instancia agradezco a la universidad UNAJMA y a mis docentes, personas de gran sabiduría, quienes se han esforzado por ayudarme, sencillo no fue el proceso, gracias a las ganas de transmitirme sus conocimientos y dedicación.

Agradezco también a mi asesor de tesis Ing. Richard, Flores por haberme brindado la oportunidad de recurrir a sus capacidad y conocimiento, así como también haberme tenido toda la paciencia del mundo para guiarme durante el desarrollo de la tesis.

Mi agradecimiento también va dirigido al Gerente Propietario de la Empresa Radial "La Primera 89.1 FM" por haberme aceptado que se realice mi tesis en su prestigiosa empresa.

Para finalizar, también agradezco a todos los que fueron mis compañeros de clase ya que gracias al compañerismo, amistad y apoyo moral tengo las ganas de seguir adelante en mi carrera profesional.

RESUMEN

El presente trabajo desarrolla un sistema web de recomendación de noticias basado en contenido que permite la búsqueda de información relevante y actualizada de las páginas web de noticias. Se implementan los módulos de gestión de usuarios, módulo de categorías, módulo de búsquedas de noticias, módulo de recomendación.

La técnica utilizada para la extracción de información es web Scraping, y se utiliza para obtener el contenido en formato RSS de las páginas de noticias. La aplicación web está desarrollada en un servidor web Django basado en Python con arquitectura MVT (modelo, vista, plantilla). El algoritmo de recomendación basado en contenido es TF-IDF que identifica ciertas palabras de un sitio web que están relacionadas con un tema específico.

El sistema web de recomendación de noticias basado en contenido se desarrolló utilizando la metodología ágil XP (programación extrema), del mismo modo se utilizó las herramientas de desarrollo como lenguaje de programación Python, servidor web Django PythonAnyware, motor de base de datos MySQL, Materialize para el diseño web, STAR UML, Framework web de modelo-vista-template de Django.

Palabras clave: Recomendación basada en contenidos, web Scraping

ABSTRACT

This work develops a content-based news recommendation web system that allows the search for relevant information and updates of news web pages. User management modules, categories module, news search module, recommendation module is implemented.

The technique used for the extraction of information is web Scraping, and it is used to obtain the content in RSS format of the news pages. The web application is developed on a Python-based Django web server with MVT architecture (model, view, template). The content-based recommendation algorithm is TF-IDF that identifies certain words on a website that are related to a specific topic.

The content-based news recommendation web system was developed using the agile XP methodology (extreme programming), in the same way, development tools such as Python programming language, Django PythonAnyware web server, MySQL database engine, materialize were used. for web design, STAR UML, Django model-view-template web framework.

Keywords: Content-based recommendation, web Scraping

CHUMASQA

Kay llamkayqa puririchinqa huk sistema web nisqantam, willarikuykuna qawanata, chaypim churamun imahinatam allinkaq, chayllaraq tarikusqa web willakuq yachaykunata maskarinanpaq. Churarimunqa imahinatam maskaqkuna, qawaqkuna tarinanpaq, imahina tarikusqankunata, imahinatam willakuykuna maskarinata, imahina qawaykachinata, ima huñusqakunata.

Chay llankaypa aypakunan willakuykuna hurqunapaqqa web Scraping kasqantakamam; yachay aypanapaqmi formato RSS willakuykunapa kaqnintakama. Aplicación web rurasqa tarikun web Django chay Python arquitectura MVT (chayhina, qawaynin, kasqan). Algoritmo qawachimusqanmi kachkan TF-IDF qawachin wakin rimaykunata chay web nisqanpi maskaspa tarinanpaq.

Chay sistema web churamun maynintam chaypa willakuyninkunaman chayanata, chaypaqmi utqayman metodología XP (hawan qawachisqa) tarimun; chaynallataqmi chay llamkana rimaykunata churamun Python, servidor web Django PythonAnywarw nisqantakama, maypi kaqkuna kallpantam MySQL, Materialize chay qawachisqa web, STAR UML, Framework web qawana template de Django chaynintakama.

Chay rimaykuna: Imahina yachaykuna kaqkuna, web Scraping

Índice

Lista de tablas

Lista de figuras

INTRODUCCIÓN

En la actualidad existe la necesidad de obtener información de forma automatizada, de acuerdo a los intereses que cada persona tiene, es decir, hacer un seguimiento a las actualizaciones de la información.

La dificultad de buscar o revisar si hay noticias de último momento o nuevos artículos, a veces resulta una pérdida de tiempo ya que estas en ocasiones no se actualizan, y si tomamos en cuenta el volumen de estos, es casi imposible de ver todos. Es por eso que se vio la necesidad de implementar una aplicación para automatizar y alertar por medio de un Smartphone, si existen noticias nuevas en los sitios de interés de una persona.

La aplicación a desarrollarse tiene como propósito ofrecer a los usuarios una aplicación web/móvil, para gestionar y notificar la existencia de información nueva en un sitio web de contenido, mediante Web Scraping. Con la aplicación de los principios y prácticas tanto en ciencias de la computación y de ingeniería de software para la creación, operación y mantenimiento de aplicaciones informáticos, el desarrollo de procedimientos y sistemas de aplicaciones que se utilizan en máquinas electrónicas y dispositivos móviles.

Estos resultados se utilizan para hacer recomendaciones de noticias basadas en el contenido, mediante el algoritmo TF-IDF, que calcula la similitud en base a las palabras contenidas en cada artículo de noticia.

Capítulo 1 PROBLEMA DE INVESTIGACION

1.1. Planteamiento del Problema

La comunicación en el siglo XXI es un gran reto, el periodismo es la "Actividad profesional que consiste en la obtención, tratamiento, interpretación y difusión de informaciones a través de cualquier medio escrito, oral, visual o grafico" (RAE-La 23a Edición, 2019a). Esta actividad es realizada por el periodista que es una persona profesional (RAE-La 23a Edición, 2019b).

En la actualidad las tecnologías de información han permitido que los medios de comunicación tradicionales migraran a medios de comunicación virtuales permitiendo mostrar gran cantidad de información de manera ordenada y en tiempo real. La información en la web sobre política, deporte, cultura, economía, educación, tecnología y otros hechos de la que queremos saber más, es imposible informar del todo, primero porque no todo es noticiable y segundo porque la información tiene que ser de interés del público. (Orts, 2020). Por ello es necesario buscar y clasificar las noticias relevantes del día.

Según el estudio de la UNESCO, menciona que "Las noticias falsas se abren paso por el mundo y representan una amenaza para el periodismo. A nivel mundial, existe la gran desinformación y noticias falsas sobre el antídoto del coronavirus, estas se extienden en internet a un ritmo aterrador. En un análisis reciente encontró que más del 40% de las publicaciones sobre el COVID-19 en una de las principales plataformas de redes sociales fueron publicados por bots-programas automatizados disfrazados de personas" ello comprobó la importancia de la información de calidad. Esta crisis de coronavirus causo "un aumento de la demanda de noticias en la web del 80 % en los últimos meses" revelo Diego Garazzi, presidente de la Comisión de Propiedad Intelectual de ADEPA, quien dialogo con una experta estadounidense, Danielle Coffey, y uno europeo, Wout Van Wijk. (Un Panel de Expertos Analizó La Importancia de Valorar

La Producción de Las Noticias, Las Fake News y El Futuro Del Periodismo - Infobae, n.d.).

Según el estudio de la Universidad De Lima, "Se calcula que para el año 2022 el noventa por ciento de la información será falsa. Parece una cifra muy alta, pero para que una noticia sea falsa no necesariamente tiene que ser inventada. Basta que tenga un adjetivo de más o un digito de menos para que se vuelva falsa". Por ello genera una gran desinformación sobre todo en sasos como el que vivimos actualmente con la pandemia del coronavirus. En el Perú, el Ministerio de Justicia y Derechos Humanos informo que "Las personas que difundan información falsa para obtener provecho o generar perjuicio a terceros será reprimido con 2 a 4 años de pena privativa de la libertad". (MinjusDH, 2020)

La falta de creación y fortalecimiento de redes de colaboración entre los periodistas nacionales, representa una dificultad para la agrupación natural de noticias con temas de interés al oyente de un determinado lugar.

Desde el inicio de su labor el productor de la Radio La Primera realiza la búsqueda de noticias internacionales, nacionales y regionales por la web para los cinco programas de la radio: primera noticia, actualidad política, agro noticias, la voz de los pueblos y talento cultural, tiene dificultades para el proceso de búsqueda, verificación y clasificación de las noticias más relevantes que existen en las diferentes páginas web de fuentes oficiales de las noticias, ya que no cuenta con un mecanismo adecuado. La difusión de noticias no verificadas y no clasificada afecta en la credibilidad de la empresa radial.

Por ello se propone desarrollar un sistema de recomendación de noticias basado en contenido para los periodistas de la radio La Primera 89.1 FM

1.2. Formulación del Problema

El proceso de Desarrollo de un sistema web de recomendación de noticias basado en contenido permitirá la búsqueda de información relevante y de actualidad de las páginas web de noticias.

1.3. Justificación

Las nuevas tecnologías de comunicación, la mayor conectividad y uso de las plataformas sociales como nuevos medios de información, han permitido un cambio de paradigma a la hora de crear, difundir y consumir noticias, en la actualidad existen miles de noticias falsa en web que afecta negativamente a la sociedad. Para no caer en esta desinformación el productor de la Radio La Primera 89.1 FM. mantiene un sistema tradicional de búsqueda de noticias debido a ello pierde demasiado tiempo en buscar y verificar la fuente de información de una noticia de interés.

Por lo cual el presente proyecto se realizará para reducir el tiempo y esfuerzo en buscar e identificar la fuente de las noticias, que realizan los periodistas de la radio La Primera 89.1 FM. El modelo web permitirá realizar una búsqueda rápida e inmediata de la noticia al mismo tiempo la verificación de la fuente de información para luego recomendar noticias relevantes al productor de la radio.

1.4. Viabilidad

1.4.1. Viabilidad Técnica

El presente proyecto a desarrollar es viable técnicamente ya que se dispone de las herramientas tecnológica necesarias para el desarrollo del proyecto. Porque el hardware o computador donde se lleve la implantación y desarrollo del software se tiene disponible y cumple con los requerimientos mínimos para su funcionamiento; en el caso del software se dispone con las aplicaciones, herramientas y algoritmos de búsqueda necesarias los cuales no ameritan inversión alguna para su adquisición también se cuenta con el servicio de internet ilimitado que amerita una inversión mínima.

1.4.3. Viabilidad Económica

Es viable económicamente porque los gastos que se emplearan para desarrollar este proyecto son en un pequeño porcentaje para ello se cuenta con los recursos económicos necesarios, es decir para la adquisición de los bienes y pago de servicios, en caso del software están disponibles en internet y es de uso gratuito. También se cuenta con los recursos financieros suficientes para pagar a los colaboradores de proyecto (recursos humanos involucrados en el desarrollo de proyecto).

1.4.4. Viabilidad Social

Este proyecto es viable socialmente porque el modelo a desarrollar será una herramienta de búsqueda y verificación de vital importancia para el medio de comunicación radio La Primera que en su día a día tiene que estar buscando noticias en la web y verificando la fuente de información el cual es tedioso y aburrido a causa de ello puede informar solo la primera noticia que le aparece en sus búsqueda sin verificar y correr el riego de que sea una falsa noticia y perder su credibilidad, no solo beneficiará a la empresa radial sino a toda la audiencia en general.

1.5. Limitaciones de la Investigación

- Solo se podrán descargar contenidos de páginas de diarios que utilicen RSS.
- El modelo a desarrollar solo funciona con acceso a internet.
- El proceso de búsqueda y análisis solo funciona para páginas web.

1.6. Objetivos

1.6.1. General

Desarrollar un sistema web de recomendación de noticias basado en contenido para medios de comunicación.

1.6.2. Específicos

- Desarrollar el modelo de búsqueda basado en contenido
- Desarrollar el módulo de recomendaciones

Capítulo 2 ANTECEDENTES

2.1. Antecedentes a Nivel Internacional

(Aldo Capello, 2018) en su informe de tesis titulado "Sistema de recomendación para textos legales", de la Universidad Nacional de Córdova, facultad de Física y Computación.

Desarrolló un sistema de recomendación de textos legales para estudiantes de derecho como para avanzados en la profesión por utilidades de aprendizaje, investigación y ayuda al exponer textos semejantes que se lee, este software estuvo basado en semejanza textual sobre el dominio de la legislación argentina, el sistema adaptado al contenido brinda recomendaciones de textos existentes en una plataforma lo cual tuvo aplicación directa a la página web de información legal de Argentina , también es adaptable a otra clase de textos, y de gran utilidad para conjuntos de textos no tan estructurados, como por ejemplo fallos judiciales, entre muchos otros, el dicho sistema de recomendación brinda la posibilidad de seguir consultado contenidos de interés sin realizar búsquedas específicas.

(Ibáñez Micó, 2015) en su informe de proyecto final de carrera titulado "Extracción de datos web usando técnica de screen-scraping", de la Universitat Politécnica de Valencia, Escola técnica superior de Informática Aplicada.

Este proyecto utilizo web Scraping para obtener información de diferentes páginas web y juntar todos los datos en una misma web generada dinámicamente. Una de sus características principales radica en hecho de que si por motivos externos a la aplicación, como por ejemplo una caída del servidor en el que esta almacenada la página sobre la cual queremos extraer la información, no podemos obtener parte de los datos que necesitamos, la aplicación funciona correctamente mostrando el resto de los valores.

(Chávez Arce, 2016) en su informe de tesis titulado "Buscador Inteligente basado en el comportamiento semántico y lenguaje natural en la Web", de la Universidad Mayor de San Andrés Facultad de Ciencias Puras y Naturales Carrera de Informática.

Se vio en la necesidad de realizar un buscador inteligente basado en el comportamiento semántico y lenguaje natural que mejore los resultados, en un proceso de búsqueda de información, que emite un usuario. Ya que en la actualidad los usuarios emiten una búsqueda de información por internet, puede observar diferentes resultados los cuales no siempre son los esperados también porque los buscadores de información actuales, no se adecuan a las necesidades de los usuarios algunos cobran dinero para satisfacer las mismas y los resultados en el proceso de búsqueda, en su mayoría son repetitivos y originan un mismo lugar, muestran el mismo sitio web como si fuera otro. Muestran masiva cantidad de información, pero de lo que no se desea, esto por derivaciones al proceso de una búsqueda de información, como por ejemplo se busca café y muestra que derivan como leche, chocolate y otros. El tiempo de búsqueda es otro factor a observar, puesto que, al buscar los resultados adecuados, esta demanda mucho tiempo.

2.2. Antecedente a Nivel Nacional

(Universidad Nacional de Piura, 2019) en su informe de tesis titulado "Desarrollo de una aplicación móvil para el acceso a información de los servicios básicos de los usuarios de la cuidad de Piura", de la Universidad Nacional de Piura, Facultad de Ingeniería Industrial, Escuela profesional de Ingeniería Informática.

Tuvo por objetivo desarrollar una aplicación móvil como medio tecnológico que facilite al usuario acceder a consultar información de los servicios básicos de luz, agua y telefonía en la cuidad de Piura, ya que los usuarios requerían acceder a consultar información como dirección de las entidades y empresas que brindan dichos servicios básicos, números de contacto telefónico, correo electrónico, horarios de atención, pasos

para hacer reclamos y realizar trámites. Utilizo la técnica web Scraping para mostrar la información de las páginas web de los servicios básicos de luz, agua y telefonía.

(Huaman & Quispe, 2019) en su informe de tesis titulado "modelo de búsqueda de productos alimenticios en supermercados online categoría abarrotes utilizando asistente virtual de tipo Chatbot y extracción de datos con Web Scraping" de la Universidad Tecnológica del Perú, facultad de ingeniería, carrera profesional de ingeniería de sistemas e Informática.

Debido a la preocupación de las tiendas online para ofrecer a su usuario una atención de calidad y personalizada, mostrando productos según sus necesidades de compra y requerimientos tecnológicos, propone un asistente virtual tipo Chatbot que cuente con la interfaz de Facebook Messenger, que beneficia al usuario para consultar los productos de los supermercados online Metro y Tottus que fue integrado con la técnica web Scraping para el almacenamiento automatizado de datos, permitiendo obtener información actualizada y real, ofreciendo la posibilidad de reducir el tiempo en la consulta de un producto, este muestra los productos disponibles de acuerdo con sus necesidades para que el usuario opte por la mejor opción.

Capítulo 3 MARCO TEÓRICO

3.1. Definición Conceptual

3.2. Sitio Web

Se conocen como colecciones de páginas web que se alojan en internet, para acceder a las páginas de un sitio web se utiliza la URL que se encuentra en un servidor, algunos requieren una suscripción para accederlas, unos nos permiten ver todo su contenido mientras otros no; los sitios de noticias proporcionan su información en tiempo real.

Se dispone de gran cantidad de herramientas para poder diseñar sitios web, independientemente de la complejidad de estos, de su rapidez, de su grado de aproximación al usuario (Cabello, 2015)

3.3. Métodos y técnicas de Recopilación de Información Web

Las tecnologías de extracción de datos e información de la Web han supuesto una revolución en el campo de la interacción usuario-ordenador. Facilitan el acceso a una cantidad impresionante de información que puede ser transformada, convirtiéndose en beneficios para los intereses del usuario final. Con el trascurso del tiempo se han desarrollado un conjunto de métodos y técnicas para la recopilación de información en la Web y su posterior uso (Ochoa, 2017). A continuación, se abordan sobre algunos métodos y técnicas utilizados para la recopilación de información en Internet:

3.3.1. Manual

Método más primitivo y básico de recopilar los datos de las páginas web, mediante el copiar y pega de los datos seleccionados. Este proceso es lento y por lo general, solo se realiza una vez.

3.3.2. Búsquedas en Internet

Método mediante el cual buscadores web, como Google.com, Yahoo.es y otros, obtienen la información asociada a un parámetro de búsqueda para su posterior recopilación.

3.3.3. Programación HTTP

Permite la creación de formas para facilitar el acceso y obtención de la información contenida en la Web. Entre sus principales usos en sitios web encontramos los RSS un formato XML para sindicar o compartir contenido en la Web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito. Este método a pesar de ser bastante utilizado, depende totalmente de la información que el sitio desee brindar o compartir y si el sitio presenta o no el uso de RSS, por lo que esta opción no cumple lo requerido para la solución (Ochoa, 2017).

3.3.4. Análisis de HTML

Esta técnica permite extraer datos mediante la detección de una secuencia de comandos, plantilla y/o código común en una página web. Esto se lleva a cabo generalmente por uno de los muchos idiomas principales de programación como XQuery, HTQL, Python, PHP, etc. (Ochoa, 2017).

3.3.5. Análisis de DOM (Modelo de Objetos de Documento)

Es la práctica de recuperar el contenido dinámico generado por los scripts del lado del cliente que se ejecutan en un navegador web. Secuencias de comandos del lado del cliente por lo general se incrustan dentro de un documento HTML o XHTML. El contenido dinámico es típicamente formateado en XML que le permite ser transferido desde el sitio web en su formato especificado.

3.3.6. Web Scraping software.

Los Software de Web Scraping normalmente simulan la forma en que los humanos exploran la web, también transforma el formato inutilizable y no legible de los datos (formato HTML) en utilizable y formato legible (contenido original como texto, imágenes, etc.), que pueden ser fácilmente exportados a hojas de cálculo para su posterior análisis (daniel Glez-Peña, 2013).

Los métodos anteriormente descritos son factibles para la obtención de información como un conjunto o un todo, pero si se desea obtener algunos datos

específicos de dicha información, se debería recurrir al método manual para su realización. Para resolver las dificultades anteriormente mencionadas surgen los métodos de Web Scraping, en español también conocidos como métodos de Raspado Web (Vargiu, 2013).

3.4. Web Scraping

Según (Hernández, Vázquez Gómez, Berdejo Rincón & Calderón Maldonado, 2015), "Es el proceso de rastreo y descarga de sitios web de información y la extracción de datos no estructurados o poco estructurados a un formato estructurado".

Del mismo modo (Glez Peña, Lourenco, López Fernández & Fdez Riverola, 2013), se define como el proceso de extraer y combinar contenidos de interés de la web de forma sistemática. En ello un robot web, imita la interacción de navegación entre los servidores web y el ser humano en un recorrido web convencional. Paso a paso, el robot accede a tantos sitios web como sea necesario, analizar su contenido para encontrar y extraer datos de interés y estructura esos contenidos como desee.

Etapas:

- Acceso al sitio: web scraping establece comunicación con el sitio web de destino a través del protocolo HTTP, un protocolo de internet basado en texto sin estado que coordina las transacciones de solicitud-respuesta entre un cliente, generalmente un navegador web, y un servidor web.
- Análisis de HTML y extracción de contenido: una vez que se recupera el documento HTML, el web scraping puede extraer el contenido de interés. Como alternativa existen bibliotecas de análisis HTML, que trabajan sobre la estructura del modelo de objetos del documento de las páginas web y el lenguaje basado en selectores como XPath y la sintaxis de selectores RSS, se recomienda que las expresiones coincidentes sean lo más generales posible para que los robots sean menos vulnerables a los cambios en el documento HTML.

- Generación de resultados: el objetivo principal es transformar los contenidos extraídos en una representación estructurada que sea adecuada para su posterior análisis y almacenamiento, aunque este paso es marginal para e web Scraping, algunas herramientas son conscientes del postprocesamineto de resultados, proporcionando estructuras de datos en memoria y soluciones basadas n texto, como cadena o archivos XML o CSV.

Según (Vargiu & Urru, 2012), aprovecho de web scraping para sugerir anuncios adecuados a una página web determina. Para ello en la publicación web, dada una página web X, la información relevante son anuncios relacionados con X y la tercera parte es X mismo, ya que mostrara los anuncios filtrados. Por tal motivo implemento un sistema web que se basa en el filtrado colaborativo y explota técnicas de scraping para analizar el contenido de la página. Al hacerlo así dada una página web X, el módulo de filtrado aprovecha la colaboración de X recuperando un subconjunto de páginas pares, el contenido de estos se analiza mediante la adopción de técnica de web Scraping, para crear un prototipo rápido de extracción implemento el sistema en Python.

3.4.1. Impedimentos para utilización de Web Scraping

Existen diferentes impedimentos que impiden o dificultan la utilización de web scraping, por un lado, aquellos factores que dificultan la realización de web scraping, y por otro, las barreras legales que a lo largo de los años se han desarrollado para evitar que se pueda realizar web scraping (Rami Essaid, 2013).

a) **Factores que dificultan el uso de aplicaciones para Web Scraping.**

Existen diversos factores que dificultan la obtención de los datos mediante el uso de herramientas destinadas al web Scraping, estos factores pueden hacer que realizar la obtención sea más complicado o incluso llegando a

impedir que se realice. Algunos de estos factores son: Ejemplos de dispositivos móviles en las que se encuentra esta tecnología:

- Códigos HTML mal estructurados, por ejemplo, aquellas páginas que fueron creadas hace tiempo y no se han adaptado, de modo que su estructura es muy simple y no es posible extraer de su etiquetado HTML información estructurada.
- Páginas que contiene sistemas de autenticación como códigos y paywalls de CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). Resulta evidente que conseguir la información mediante la programación de un scraper es imposible, ya que para acceder a la página es necesario realizar una autenticación de CAPTCHA que es totalmente diferente en cada petición.
- Bloqueo al acceso masivo por los administradores de los servidores. Al programar un scraper se realizan muchas peticiones a un servidor, el cual puede detectarlo como un ataque de denegación de servicio y bloquear nuestra IP.
- Sistemas que usan cookies para realizar un seguimiento de lo que el usuario realiza.

b) **Barreras legales que limitan el Web Scraping.**

Existen limitaciones relacionadas con las barreras legales, ya que, algunos países reconocen los derechos de bases de datos limitando el derecho a reutilizar la información que se obtiene de lugares ya publicados. De hecho, algunas páginas se protegen del Scraping declarando en sus condiciones legales la prohibición de realizar Scraping sobre la página.

3.5. Sistemas de recomendación

En nuestro día a día se nos presentan situaciones en la que debemos seleccionar una opción de muchas disponibles. Tareas como elegir una noticia relevante, una

película para ver o repositorio para sacar información es dificultoso cuando la cantidad de alternativas de la que disponemos es muy grande. Para solucionar este problema surgen los sistemas de recomendación. (Proyecto, 2015). Entonces los sistemas de recomendación estudian las preferencias de los usuarios con el objetivo de sugerirles contenidos factibles y de interés. En otras palabras, es un sistema inteligente que ofrece al usuario recomendación personalizada sobre un determinado contenido (Big Data, 2018).

Según(Ml Y Bayes.Pdf, n.d.), un sistema de recomendación "es una herramienta que establece un conjunto de criterios y valoraciones sobre los datos de los usuarios para realizar predicciones sobre recomendaciones de elementos que puedan ser de utilidad o valor para el usuario estos sistemas seleccionan datos proporcionados por el usuario de forma directa o indirecta, y procede a analizar y procesar información del historial del usuario para transformar estos datos en conocimiento de recomendación". Agrega que en la actualidad los sistemas de recomendación tienen un nivel de eficiencia alto ya que puede asociar elementos de nuestros perfiles de consumo como el historial de compras, selección de contenidos y nuestras horas de actividad, para realizar las recomendaciones. A continuación, se irán definiendo con más detalle los sistemas de recomendación.

Según (Burke, 2002), para entender mejor como son los sistemas de recomendación observaremos las diferencias entre un sistema de búsqueda de información simple y como puede ser un buscador en un sitio web, y un sistema de filtrado y recomendación.

- Frecuencia de uso: Los sistemas de búsqueda son enfocados por uso puntual del usuario mientras que los sistemas de recomendación están diseñados para uso prolongado y repetitivo.
- Representación de las necesidades de información: en los sistemas de recomendación la información es descrita en los perfiles del usuario

mientras que en los sistemas de búsqueda la información requerida se expresa en forma de pregunta.

- Objetivo: Los sistemas de búsqueda seleccionan ítem de la base de datos que coincide con la pregunta mientras que los sistemas de recomendación eliminan la información irrelevante de flujos de entrada de información o reúne información relevante de diferentes repositorios de acuerdo al perfil del usuario.
- Base de datos: Los sistemas de búsqueda trabajan con bases de datos estáticas mientras que los sistemas de recomendación trabajan con información dinámica.
- Tipo de usuario: Los sistemas de búsqueda no tienen información sobre el usuario que lo utiliza mientras que en los sistemas de recomendación es necesario saber o tener información del usuario.
- Ámbito social: los sistemas de recomendación están interesados en aspectos sociales de modelado y privacidad del usuario mientras que los sistemas de búsqueda no.
- Acción: el proceso de recomendación se relaciona con la acción de "eliminar" información, mientras que el proceso de búsqueda se relaciona con la acción de "encontrar" información.

3.5.1. Clasificación de sistemas de recomendación

Según (Proyecto, 2015) del mismo modo existen los sistemas de recomendación en contenido que tiene como objetivo sugerir elemento que sean del interés de los usuarios, basado en su perfil o en su entorno social del usuario ejemplo, Amazon. También los sistemas basados en filtrado colaborativo. Así mismo existen otros SR.

a) **Sistema de recomendación basado en contenido:**

Los sistemas de recomendación basados en contenido buscan productos cuyas características se ajustan lo mejor posible a las preferencias del usuario.

la información obtenida se analiza para buscar propiedades, características y cualidades del producto. El razonamiento de estos sistemas se basa en que si a un usuario le gustan los productos de una determina página web, también le gustaran otros productos de características similares. "Los sistemas de recomendación basados en contenido poseen un gran paralelismo con el razonamiento basado en casos" (Pazzani & Billsus, 2007).

(D.Sergio, 2012), asume que los sistemas de recomendación basado en contenido proporcionan recomendaciones basadas en una relación de "articulo a artículo", siendo muy "predictivo", se basa en la información del sistema y no des usuario real. Por ejemplo, un SR de libros que aplique este tipo de sistema, los usuarios reciban recomendaciones de aquellos libros que tengan relación de contenido, con los libros accedidos previamente.

b) **Sistemas de recomendación de filtrado colaborativo:**

Utilizan la información proporcionada por el usuario de características similares al producto para generar recomendaciones, se basa exclusivamente en las valoraciones que reciben por los usuarios, cuando se agrupan a los usuarios en categorías específicas. El sistema tiene a ofrecer resultados pobres si se cuenta con poca información sobre los usuarios (E.Peis;J.M.Morales del Castillo;A.Desgado López, 2008)

c) **Sistemas de recomendación basado en conocimiento:**

Aunque los SR mencionados anteriormente son los más conocidos, pero no en todas las situaciones son los más adecuados. Entonces los SR basado en conocimiento, utilizan el conocimiento que proporciona el usuario sobre sus necesidades y el conocimiento que tiene el sistema sobre los productos para realizar recomendaciones de los productos que mejor cubre las necesidades de los usuarios (Barranco et al., 2006).

3.6. Mesología Ágil

Según (Montero Molina et al., 2018), en la actualidad el software es participe casi todas las operaciones empresariales por ende se debe desarrollar soluciones informáticas de manera ágil para poder dar una respuesta de calidad a todo lo necesario.

Las metodologías agiles presentan como principal particularidad la flexibilidad, los proyectos en desarrollo son subdivididos en proyectos más pequeños, incluye una continuación constante con el usuario, es altamente colaborativos y es mucho más adaptable a los cambios. De hecho, el cambio de requerimiento por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constante; entre las más notables metodologías de desarrollo ágil se encuentran Scrum (muy popular en emprendimientos), Programación Extrema, Móvil-D (ágil extremo para móviles).

3.6.1. Scrum

Según (Perez, 2011), No corresponde a ningún acrónimo, su nombre proviene del deporte rugby, que es una formación requerida para la recuperación rápida del juego ante una infracción menor.

Scrum es un marco de trabajo diseñado de tal forma que logra la colaboración eficaz del equipo de trabajo, emplea un conjunto de reglas y se definen roles para generar una estructura de correcto funcionamiento.

Según (Cevallos, 2018), Scrum define tres roles, los cuales son: El Scrum master, el dueño del producto y el equipo de desarrollo. El scrum master es la persona que lidera el equipo asegurándose que el equipo cumpla las reglas y procesos de la metodología. El dueño del producto es el representante de los accionistas y clientes que usan el software. El equipo de desarrollo es el grupo de profesionales encargados de convertir la lista de requerimientos o también llamado Product Backlog en funcionalidades del software.

Scrum utiliza un elemento representativo llamado Sprint, que corresponde a una etapa de trabajo donde se crea una versión utilizable del producto. Cada sprint es

considerado como un proyecto individual. Un Sprint está compuesto por los siguientes elementos: reunión de planeación del Sprint, Daily Scrum o reunión diaria, trabajo de desarrollo, revisión y retrospectiva del Sprint (Montero, 2018).

3.6.2. XP (Programación Extrema)

La metodología entreme programming o XP, es la metodología ágil más conocida (Sato, Bassi, Golman & Kon, 2009). Fue desarrollada por Kent Beck, en la búsqueda por guiar equipos de trabajo pequeños o medianos, en ambientes de requerimientos imprecisos o cambiantes.

La principal particularidad de esta metodología son las historias de usuario, las cuales corresponden a una técnica de especificación de requisitos; se trata de formatos en los cuales el cliente describe las características y funcionalidades que el sistema debe poseer (Albetoc, 2015).

Así mismo, "se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en la soluciones implementadas y coraje para enfrentar los cambios "(Letelier & Penadés, 2012).

3.6.2.1. Fase de Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

3.6.2.2. Fase de Planificación

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

3.6.2.3. Fase de Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fuercen la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción.

3.6.2.4. Fase de producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

3.6.2.5. Fase de Mantenimiento

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Capítulo 4 MATODOLOGÍA DE LA INVESTIGACIÓN

4.1. Análisis de contenido

El sistema de recomendación basado en contenidos trabaja con datos que el usuario proporciona, directamente mediante una calificación o implícitamente por hacer click en un link. En base a los datos, se genera un perfil del usuario, el cual se utiliza para generar sugerencias al usuario. Conforme el usuario proporciona más entradas o tomar acciones en estas recomendaciones, el motor llega a ser poco a poco más exacto.

Un sistema de recomendación tiene que decidir entre dos tipos de métodos para la información que se entrega al usuario con recomendaciones:

MÉTODOS	DESCRIPCIÓN
Explotación	El sistema elige documentos similares a aquellos que el usuario ha expresado sus preferencias
Exploración	el sistema elige documentos cuando el perfil del usuario no proporciona evidencia para predecir la reacción del usuario

Tabla 1 Tipos de métodos

Fuente: Elaboración propia

4.1.1. Algoritmo TF-IDF

Según (Maarif, 2015), se utiliza para darle peso a una palabra clave en un documento y asignarle una importancia a la palabra en base a la cantidad de veces que aparece en el documento. El peso más alto se le asigna a la palabra que aparece más veces. Cada palabra tiene su respectivo score TF e IDF. El producto de los escores TF e IDF se llama peso TF * IDF.

El TF (Term Frecuency) de una palabra es la cantidad de veces que aparee en un documento. Conociendo este, es posible que veas que estás usando un término más frecuentemente.

El IDF (Inverse Document Frequency) es la medida de cuan significativo es un término en el corpus.

$$W_{x,y} = tf_{x,y} \times \log \frac{N}{df_x}$$

En el programa se utiliza la biblioteca scikit-learn que proporciona un vectorizador TF-IDF que calcula el puntaje TF-IDF para cada descripción de noticia, palabra por palabra.

4.1.2. Modelo espacio vectorial

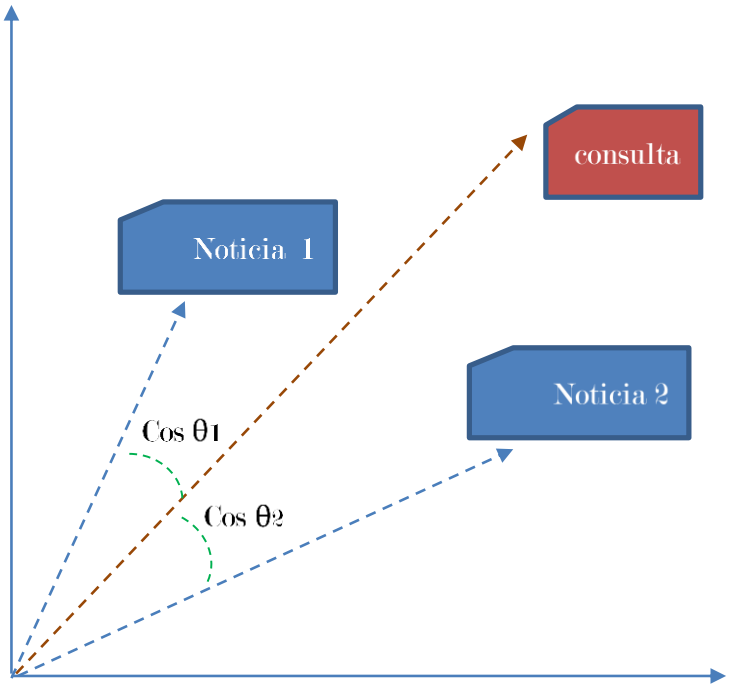


Figura 1 Noticias representadas como vectores

Fuente: Elaboración propia

En este modelo, cada ítem se almacena en un vector de sus atributos (el cual también es un vector) en un espacio n-dimensional, y se calculan los ángulos entre los vectores para determinar la similitud entre vectores.

El método de calcular la medida del gusto del usuario se calcula tomando el coseno del ángulo entre el vector del perfil del usuario y el vector del documento, o en nuestro caso, el ángulo entre dos vectores de noticias.

La razón de utilizar el coseno es que el valor del coseno se incrementa cuando el ángulo entre los vectores decrece, lo que significa mayor similitud.

Los vectores luego se normalizan en longitud, después de esto llegan a ser vectores de longitud 1. Este ejemplo muestra cómo se calcula.

Contenido 1: El carro es conducido en la carretera

Contenido 2: El camión es conducido en la autopista

En este ejemplo cada contenido es un documento separado Calculando el TF-IDF para los documentos anteriores, los cuales representan nuestro corpus.

Palabra	TF		IDF		TF*IDF	
	A	B			A	B
EL	1/7	1/7	Log (2/2)	0	0	0
CARRO	1/7	0	Log (2/1)	0.3	0.043	0
CAMION	0	1/7	Log (2/1)	0.3	0	0.043
ES	1/7	1/7	Log (2/2)	0	0	0
CONDUCIDO	1/7	1/7	Log (2/2)	0	0	0
EN	1/7	1/7	Log (2/2)	0	0	0
LA	1/7	1/7	Log (2/2)	0	0.043	0
CARRETERA	1/7	0	Log (2/1)	0.3	0	0.043
AUTOPISTA	0	1/7	Log (2/1)	0.3	0	0

Tabla 2 Ejemplo de cálculo de TF-IDF

Fuente: Elaboración propia

4.2. RSS

El Real Simple Sindicación (RSS) es un formato de publicación de información basado en XML, donde se especifican los datos fundamentales de los ítems informativos que, de forma paralela, se han publicado en un sitio Web. Es decir, un sitio Web publica noticias, artículos u otro contenido y de forma paralela publica un fichero RSS que contiene los datos básicos y los resúmenes de aquellas informaciones, incluyendo para

cada una de ellas un enlace que permita al lector del RSS localizar la información original (Miguel Angel, Mayer Pujadas, Luis Garcia, Angela Leis, 2008).

En estos momentos un gran porciento de los sitios en Internet de publicación de noticias, artículos científicos, etc., hacen un uso intensivo de la publicación mediante RSS, como forma de lograr una mayor difusión de sus contenidos. Cualquier sitio Web que publique nuevos contenidos con una cierta regularidad es un candidato perfecto para emplear los RSS.

Por lo general, los canales RSS se identifican en los sitios Web que los contienen por un icono de color naranja, que contiene las iniciales RSS o XML.

Este formato ha sido diseñado para contener la información básica de los ítems informativos que se incluyen en él de manera que esta información pueda ser procesada por ordenadores. De esta manera, al obtener un fichero RSS desde Internet, una computadora puede realizar cualquier procesamiento que pueda programarse sobre dicha información, incluyendo definición de palabras claves, categorización, mostrarla visualmente con un formato nuevo preconcebido, incluirla en boletines electrónicos, etc.

De ahí la utilidad de esta técnica, que permite a los usuarios apoyarse en diversas herramientas que ayuden en el procesamiento de la información, sin que exista la necesidad de navegar por distintos sitios Web para obtener la información que se desea. Basta con "suscribirse" a los canales RSS deseados para recibir de forma automática la información que aquellos contienen. Es importante destacar el hecho de que el RSS no fue diseñado para ser leído por las personas. Aunque puede ser comprendido, para esto es necesario un cierto nivel de conocimientos técnicos sobre su naturaleza.

También (Sanches, 2007), afirma que existen varias versiones del formato RSS, algunas de las cuales permiten incluir marcadores de información personalizados para

aquellos casos en que sean necesarios. Normalmente se usan dos formatos principales en la publicación de contenido en Internet, que son el RSS 0.9x y el RSS 1.0. Existen también módulos adicionales para la personalización de estos formatos.

4.2.1. Obtención de información con canales RSS

(Sanches, 2007), lo primero para poder obtener nuevos contenidos con la ayuda de canales RSS es localizar los canales que son de nuestro interés. Esto puede lograrse directamente, al visitar un sitio en Internet cuyo contenido nos resulte útil, buscando si dicho sitio publica en formato RSS. Otra vía es visitando directorios de canales RSS y localizando allí aquellos que nos pudieran interesar. Finalmente, se pueden localizar también sencillamente haciendo una búsqueda en Internet sobre el tema de interés y adicionando a las palabras claves que se especifiquen el término "RSS". En el caso de Google, que es el directorio más visitado en Internet, basta con especificar la palabra de búsqueda y adicionar el término "filetype:rss" para indicarle que en el resultado de la búsqueda se quiere obtener exclusivamente las direcciones de ficheros RSS.

Una vez localizados aquellos canales RSS de interés, se deben registrar en un programa conocido como "agregador". Los agregadores son aplicaciones que se encargan de conectarse con los canales RSS que se las ha especificado y obtener la información para mostrarla al usuario. En general presentan funcionalidades para facilitar su empleo, como puede ser unir contenidos procedentes de distintos canales, ocultar información que ya el usuario ha visto anteriormente y catalogar los canales y los contenidos.

4.3. Pythonanywhere

(Funes & Dasso, 2014), es una empresa de alojamiento web, que ofrece una solución de alojamiento para aplicaciones web Python, en efecto aloja, ejecuta, codifica

Python en la nube, también dispone de muchísimos repositorios libre solo hay que seguir su código en GitHub (https://www.pythonanywhere.com/).

4.4. Framework Django

(Jeff & Paul Bissex, 2008), Django es un Framework de aplicaciones web de código libre y abierto, escrito en Python, que sigue el patrón de arquitectura MVT (Model View Template) para el lenguaje de programación Python. Es mantenido por Django Software Foundation (DSF), una organización independiente, establecida como sin fines de lucro. Las bibliotecas adicionales a utilizar son: Feedparser, Pandas, Scikitlean.

4.5. Arquitectura MVT

(Martínez Acosta, 2014), MVT (Model View Template o Modelo Vista Plantilla), toma una vista de la petición HTTP como parámetro y devuelve una respuesta HTTP. Carga los datos de la vista de los modelos y la pasa a la plantilla. La plantilla controla cómo se muestran los datos. MVT se salta la estructura del controlador base y simplemente invoca una función basada en la ruta. Es más sencillo de entender y fácil de probar. En lugar de tener que imitar la construcción del objeto controlador y la invocación de la acción, simplemente puede llamar a la función de vista con un parámetro de la petición HTTP estándar.

4.6. Aplicación de la metodología XP

El desarrollo del presente proyecto, se realizará con la metodología de desarrollo XP (Programación Extrema), siguiendo las siguientes fases: exploración, planificación, iteraciones, producción y mantenimiento.

4.6.1. Fase de exploración

La fase de exploración nos permitió identificar algunos requerimientos funcionales y no funcionales que el sistema de recomendación de noticias debe cumplir.

4.6.1.1. Requerimientos funcionales

Los requerimientos funcionales del sistema describen el comportamiento o funciones del software, que se listan y detallan a continuación.

Requerimiento	Descripción
RF.01	Registro de usuarios
RF.02	Login de usuarios
RF.03	Obtención de noticias por categorías
RF.04	Búsqueda de noticias por palabras
RF.05	Presentación de noticias por categorías
RF.06	Recomendación de noticias
RF.07	Administración de usuarios
RF.08	Administración de categorías

Tabla 3 Requerimientos funcionales

Fuente: Elaboración propia

Caso de uso

Aquí se describirá las actividades que deberá realizar el sistema de software con relación a sus actores, iniciará con el actor principal que es el usuario.

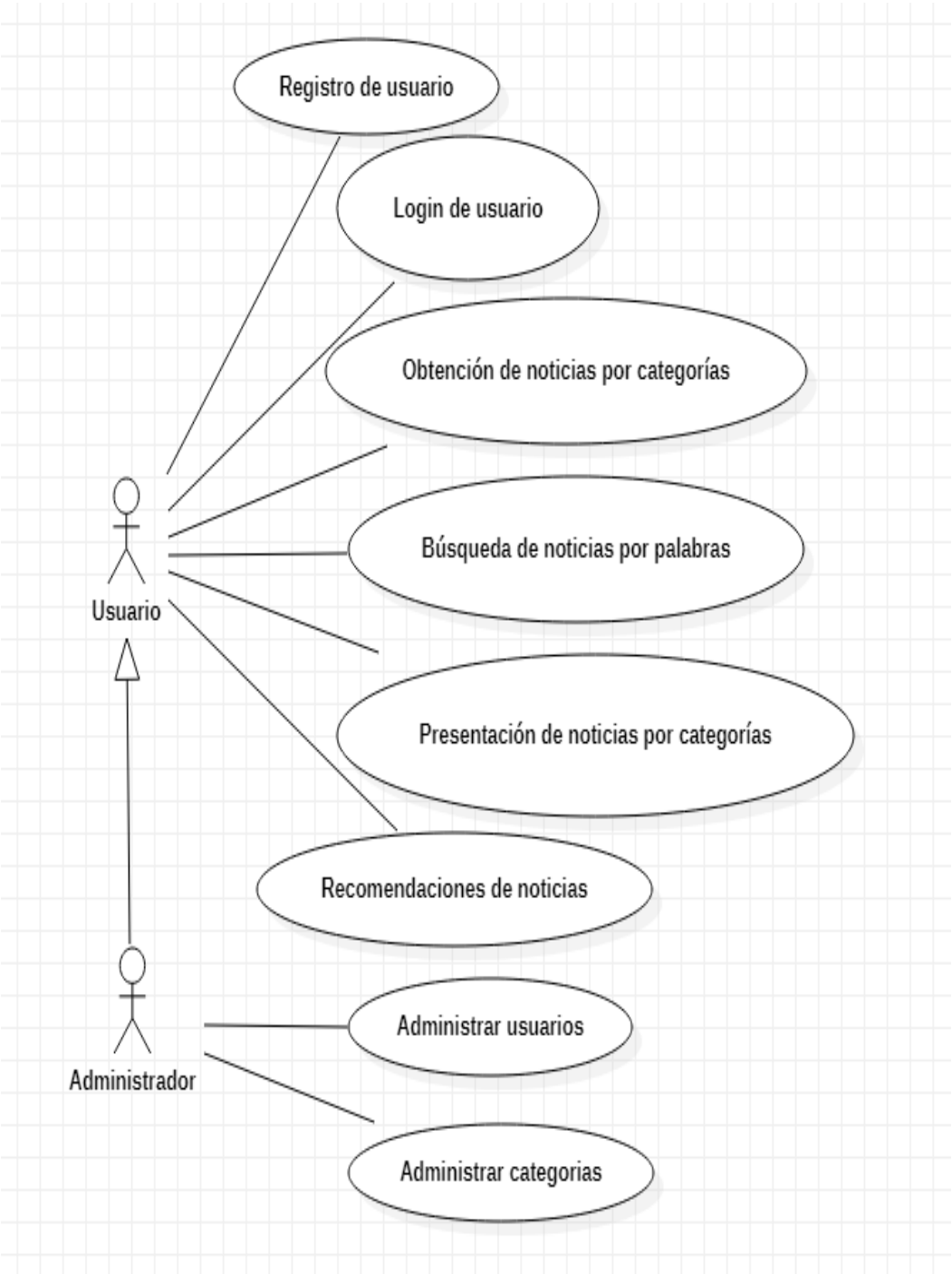


Figura 2 Diagrama de caso de uso del software

Fuente: Elaboración propia

- **Historias de usuario**

Las historias de usuario nos ayudarán de una forma rápida administrar los requisitos de los usuarios y se irán implantando en las iteraciones del proceso de desarrollo (Sommerville, 2005).

Identificador: 001	**Nombre:** Registro de usuario
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	Login de usuario
Prioridad de desarrollo	Media
¿Es crítico?	Si
Entrada	Nombre de usuario, contraseña
Salida	Mensaje de aceptación
Precondición: El usuario no está registrado en el sistema **Descripción:** El usuario ingresa los datos con los que se va a registrar su nombre de usuario y su contraseña repetida para comprobar que no la ha errado. **Postcondición:** El usuario y contraseña se registra en la base de datos y se le presenta le da acceso inmediatamente con esas credenciales	
Manejo de situaciones anormales: Si no se cumplen las restricciones de nombre de usuario y contraseña no se permite el registro	
Criterios de aceptación:	

Tabla 4 Historia de usuario 001

Fuente: Elaboración propia

Identificador: **002**	Nombre: **Login de usuario**
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	Recomendaciones
Prioridad de desarrollo	Media
¿Es crítico?	Si
Entrada	Nombre de usuario, contraseña
Salida	Mensaje de aceptación

Precondición: El usuario no está registrado en el sistema **Descripción:** El usuario ingresa los datos con los que se a registrar su nombre de usuario y su contraseña repetida para comprobar que no la ha errado. **Postcondición:** Se le concede acceso inmediatamente
Manejo de situaciones anormales: Si no consigue loguearse en 3 intentos se impiden los siguientes intentos hasta pasado un tiempo
Criterios de aceptación:

Tabla 5 Historia de usuario 002

Fuente: Elaboración propia

Identificador: 003	**Nombre:** Obtención de noticias por categorías
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	Recomendaciones
Prioridad de desarrollo	Alta
¿Es crítico?	Si
Entrada	Lista URLs de archivos rss en xml de fuentes de noticias
Salida	Estructura de datos JSON conteniendo el título, fecha, contenido, fuente del artículo obtenido
Precondición: No se han obtenido aún los xml de contenido **Descripción:** se itera sobre cada URL proporcionada obteniéndose del xml del feed de contenido en una estructura de diccionario donde cada par clave:valor corresponde a los datos de cada noticia **Postcondición:** Listado de contenidos de noticias obtenido	
Manejo de situaciones anormales: Si no se consigue acceso a una URL se devuelve un mensaje de advertencia y se procede con las demás URLs	
Criterios de aceptación:	

Tabla 6 Historia de usuario 003

Fuente: Elaboración propia

Identificador: 004	Nombre: Búsqueda de noticias por palabras
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	Login de usuario
Prioridad de desarrollo	baja
¿Es crítico?	No
Entrada	Palabra buscada
Salida	Mensaje de aceptación
Precondición: El sistema tiene las noticias cargadas desde sus fuentes **Descripción:** Se hace la consulta para la palabra buscada independientemente de la categoría a la que pertenece **Postcondición:** Se muestran solamente las noticias que contienen la palabra buscada	
Manejo de situaciones anormales: Si no se encuentra ninguna noticia que coincida con la palabra se muestra un mensaje de advertencia	
Criterios de aceptación:	

Tabla 7 Historia de usuario 004

Fuente: Elaboración propia

Identificador: 005	Nombre: Presentación de noticias por categorías
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	búsqueda de noticias
Prioridad de desarrollo	Alta
¿Es crítico?	Si
Entrada	Lista de entradas de noticias
Salida	Mensaje de aceptación
Precondición: Las noticias ya están cargadas en el programa **Descripción:** Las noticias se descargan ya categorizadas desde las fuentes de información **Postcondición:** Noticias formateadas y presentadas por grupos	
Manejo de situaciones anormales: Si alguna categoría no contiene ítems no se muestra	

Criterios de aceptación:

Tabla 8 Historia de usuario 005

Fuente:1 Elaboración propia

Identificador: 006	**Nombre:** Recomendaciones de noticias
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	Recomendación de noticias
Prioridad de desarrollo	Media
¿Es crítico?	Si
Entrada	Dataset de preferencias anteriores de cada usuario.
Salida	Lista de noticias recomendadas en base al score alcanzado
Precondición: El usuario tiene que haber elegido algunas noticias como preferidas sino se mostrarán inicialmente al azar **Descripción:** Se realiza el cálculo del score en base al algoritmo **Postcondición:** Noticias clasificadas	
Manejo de situaciones anormales: Si no se puede definir la preferencia se mostrará al azar	
Criterios de aceptación:	

Tabla 9 Historia de usuario 006

Fuente:2 Elaboración propia

Identificador: 007	**Nombre:** Administración de Usuarios
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	Administración
Prioridad de desarrollo	Media
¿Es crítico?	No
Entrada	Base de datos de usuarios
Salida	Usuarios
Precondición: Debe existir una cuenta de super usuario, esta se crea desde la consola de Django.	

Descripción: El usuario administrador puede validar o borrar usuarios registrados **Postcondición:** Lista de usuarios procesadas
Manejo de situaciones anormales: Si no se puede acceder se debe indicar que se necesita crear un usuario administrador
Criterios de aceptación:

Tabla 10 Historia de usuario 007

Fuente: Elaboración propia

Identificador: 008	**Nombre:** Administración de categoría
Tipo(necesario/deseable)	Necesario
Requerimiento que lo utiliza	Administración
Prioridad de desarrollo	Media
¿Es crítico?	No
Entrada	Base de datos de categorías
Salida	categorías
Precondición: Debe existir una cuenta de super usuario, esta se crea desde la consola de Django. **Descripción:** El administrador puede agregar o quitar categorías creadas **Postcondición:** Lista de categorías administradas	
Manejo de situaciones anormales: Si no se puede acceder se debe indicar que se necesita crear un usuario administrador	
Criterios de aceptación:	

Tabla 11 Historia de usuario 008

Fuente:3 Elaboración propia

4.6.1.2. Requerimientos no funcionales

Son los requisitos que no se refieren directamente a las funciones específicas del sistema, si no para juzgar la operación de un sistema, a continuación, se menciona. (Sommerville, 2005)

- **Requerimientos del software**

Los requerimientos del producto software que fueron considerados son los que mencionamos a continuación:

Requerimiento	Descripción
R.1	El sistema se desarrolla en plataforma web con HTML5 Y jQuery
R.2	Debe tener un diseño responsivo adaptable a dispositivos móbiles
R.3	El servidor debe correr en leguaje Python del lado del servidor
R.4	La interface web debe seguir el diseño Material de Google
R.5	La metodología de desarrollo es evolutiva y enmarcado en XP
R.6	El sistema obtendrá datos en formato XML mediante el lenguaje RSS

Tabla 12 Requerimientos del producto software

Fuente:4 Elaboración propia

- ✓ **Funcionalidad**

  La funcionalidad del software va satisfacer las necesidades de los periodistas al ser utilizados bajo condiciones específicas (Pressman, 2002).

- ✓ **Usabilidad**

  La usabilidad del software será la capacidad que va tener para ser entendido y aprendido por los periodistas cuando lo usen bajo determinadas condiciones (Pressman, 2002).

- ✓ **Mantenibilidad**

La mantenibilidad del software va tener la capacidad de ser modificado efectiva y eficientemente debido a las necesidades evolutivas (Pressman, 2002).

- ✓ **Portabilidad**

  La portabilidad del software será la capacidad que va tener para ser transferido de forma eficiente y efectiva (Pressman, 2002).

4.6.2. Fase de planificación

Aquí describimos la planificación que se ha seguido en el proyecto a lo largo de su desarrollo, para ello se tiene los siguientes módulos.

4.6.2.1. Módulo de gestión de usuario

Como todo sistema de información, nuestro software, en este módulo permite la gestión de acceso concedido a los usuarios, mencionaremos los requerimientos que se van a desarrollaran.

ID	DESCRIPCION
usuario	Se realiza el proceso de registro de un nuevo usuario
	Realiza el proceso de validación del usuario

Tabla 13 Módulo usuario

Fuente: Elaboración propia

- **Tipos de usuarios**

Los usuarios del sistema son de tres tipos: Usuarios administradores solo para tareas de depuración, pruebas y validación; Usuarios lectores registrados; Usuarios lectores anónimos.

USUARIO	DECRIPCION
Administrador	✓ Administra la tabla de datos ✓ Borra o valida a otros usuarios administradores lectores ✓ Modifica y agrega categorías ✓ Valida entradas de noticias
Lector	✓ Modifica su información personal ✓ Marcar las entradas de noticias que les gusta
Anónimo	✓ Solo poder leer las noticias

Tabla 14 Tipos de usuario

Fuente:5 Elaboración propia

4.6.2.2. Módulo categoría

En este módulo el software, muestra las diferentes funcionalidades que realiza con respecto a las noticias.

ID	DESCRIPCION
	Muestra las noticias de acuerdo a la categoría seleccionada

CATEGORIA	Ordena las noticias por publicaciones recientes
	Muestra el contenido de la noticia seleccionada
	Registra las noticias de acuerdo a las preferencias del usuario

Tabla 15 Modulo categoria

Fuente: Elaboración propia

Primero se realiza el proceso de configuración del entorno de la plataforma donde funcionara el sistema web, que es Pythonanywhere para ello se crea una cuenta.

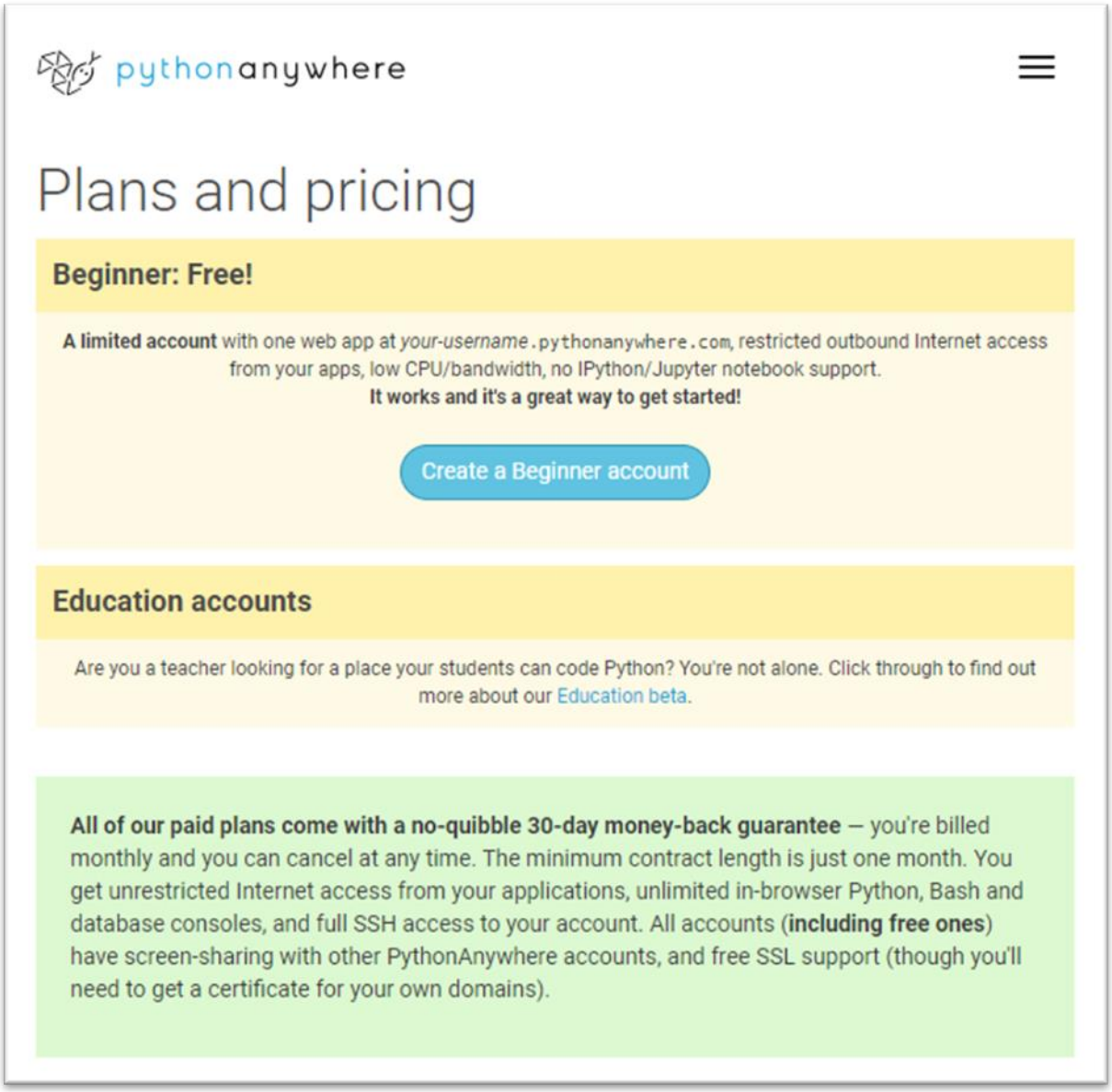


Figura 3 Servidor Python

Fuente: https://www.pythonanywhere.com/).

Del lado del servidor se utiliza el Framework Django con el lenguaje de programación Python. El que hace las funciones del servidor a continuación, la instalación de Django3.

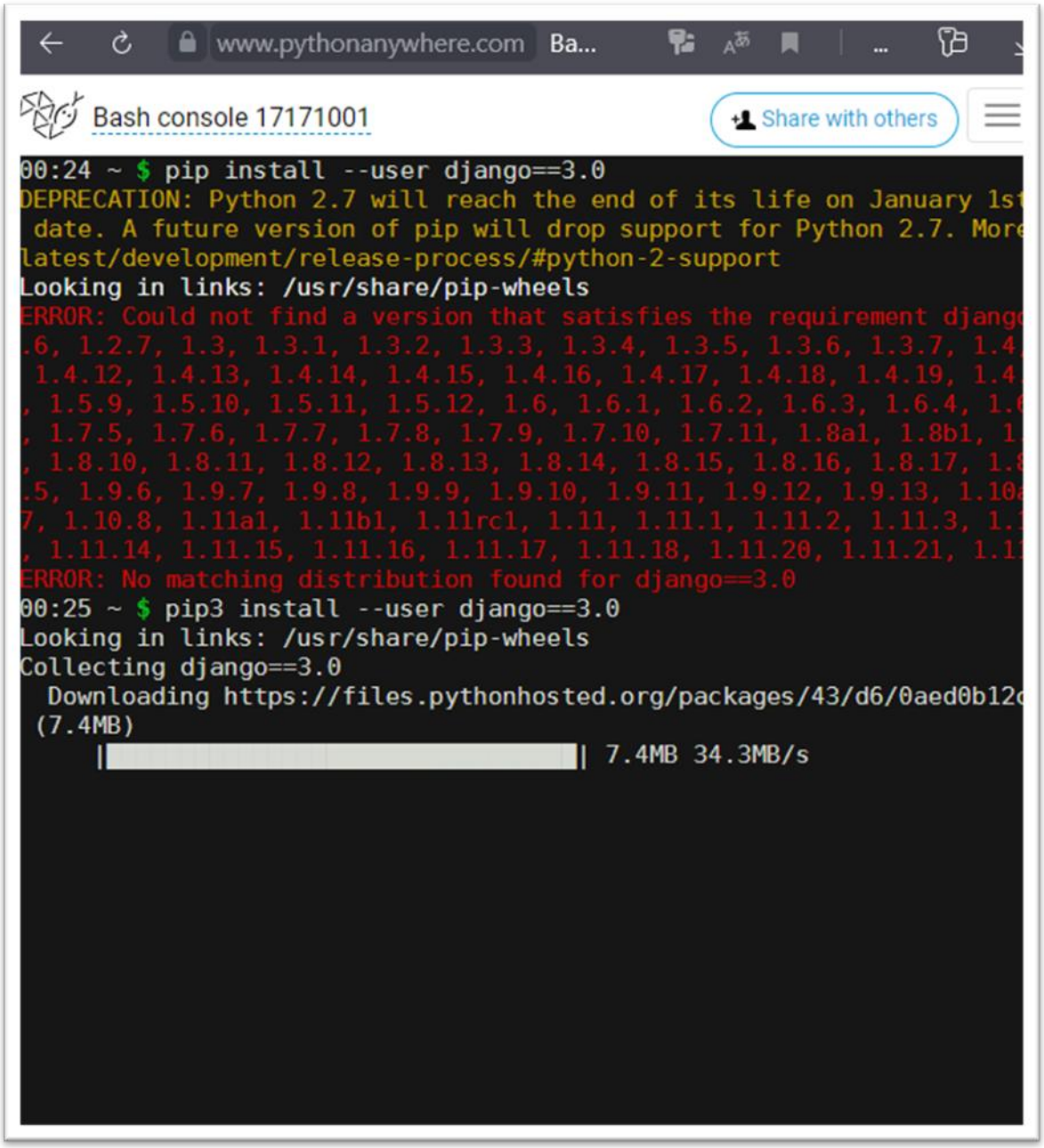


Figura 4 Configuracion del servidor Django

Fuente:6 Elaboración propia

Luego se realizará la configuración para nuestro Sistema de software.

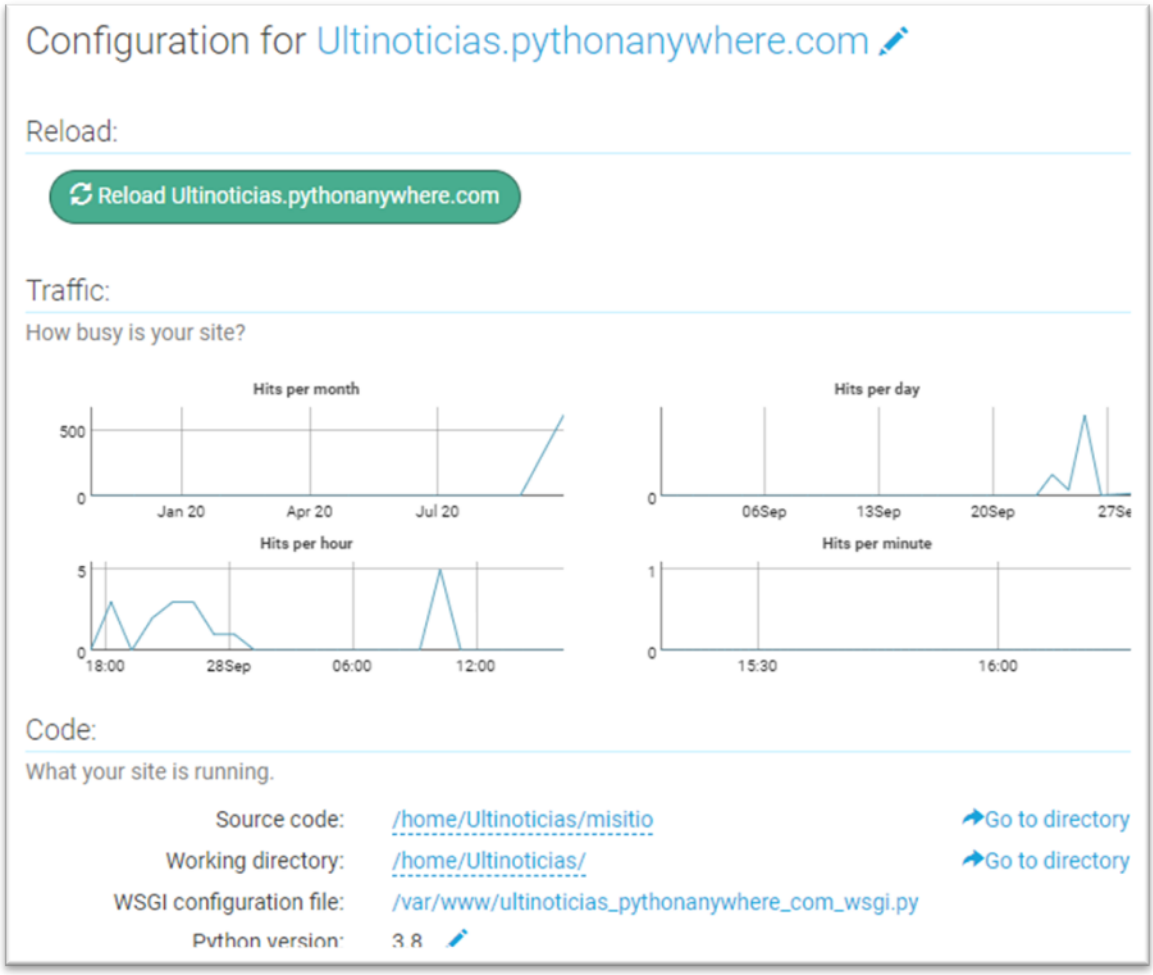


Figura 5 Servidor del sitio Ultinoticias.com en operación

Fuente: Elaboración propia

Búsqueda basada en contenido

Es el método que se utiliza internamente para recopilar la información en la web y mostrarlo en cada una de las categorías, para ello se sigue los siguientes pasos.

Recopilación

Es el proceso de identificación de las nuevas publicaciones en los sitios web y la obtención de los datos, haciendo referencia a: título, descripción, autor, url, fecha, fuente de origen, y cuerpo de la publicación; de los cuales se obtienen: el autor, la fecha de realización y el texto del mismo.

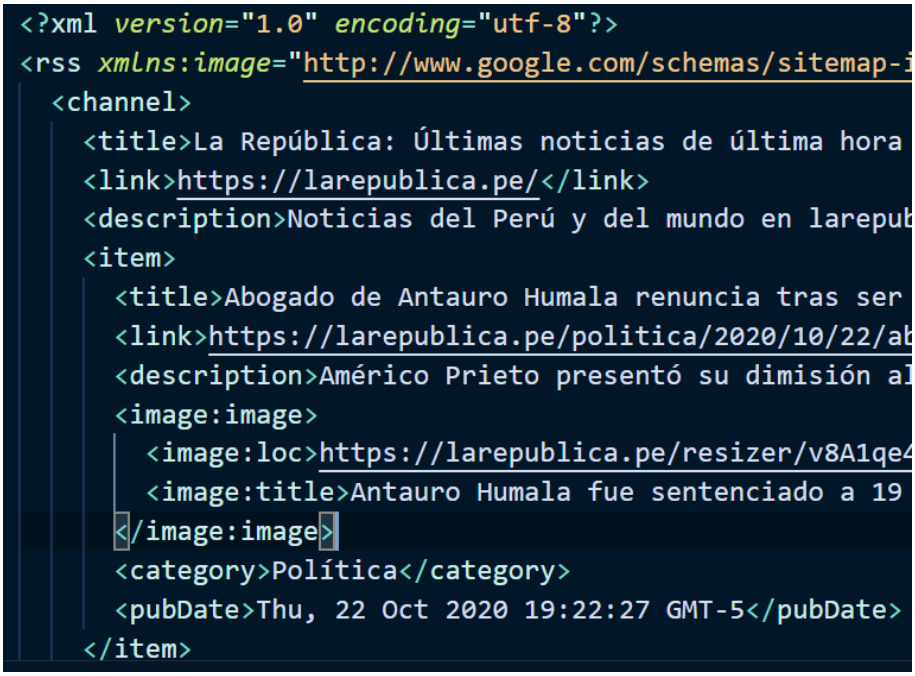


Figura 6 Estructura de un RSS

Fuente: Elaboración propia

‹title› Es el título del ítem.

‹link› La dirección en Internet de la información original.

‹description› Una descripción o resumen de la información publicada. Normalmente puede ser un resumen de una noticia o artículo.

‹pubDate› La fecha de publicación de la información original.

Estos marcadores pueden variar en cierta medida entre un formato RSS y otro, pero existe un conjunto de estándar que es utilizado en todos los formatos y que puede utilizarse para lograr una cierta compatibilidad cuando se trata de procesar un gran número de fuentes RSS.

Actualización

Corresponde a la actualización de los datos recopilados anteriormente de las publicaciones. Se actualiza la cantidad de visitas, la cantidad de comentarios, se agregan los comentarios nuevos y algún cambio en el cuerpo de la publicación.

4.6.2.3. Modulo buscar noticias

Se realiza las búsquedas de las noticias de acuerdo al interés del usuario invitado o del usuario registrado.

ID	DESCRIPCION
BUSCAR NOTICIAS	Realiza el proceso de búsqueda de noticias por categoría
	Realiza el proceso de búsqueda de noticias por palabra

Tabla 16 Módulo buscar Noticia

Fuente: Elaboración propia

4.6.2.4. Módulo de recomendación

Muestra un resumen sobre la información recopilada y/o actualizada para cada tipo de usuario que interactuó con el software, de acuerdo a sus preferencias.

ID	DESCRIPCION
RECOMENDACION	Realiza el proceso de búsqueda de noticias por categoría
	Realiza el proceso de búsqueda de noticias por palabra

Tabla 17 Módulo recomendacion

Fuente:7 Elaboración propia

4.6.3. Fase de iteraciones

La metodología XP implica la realización de iteraciones, en cada una de estas el producto software fue implementando nuevas características en base a las prioridades definidas en los requerimientos funcionales y otros que fueron surgiendo en el desarrollo del mismo. Las iteraciones realizadas son:

4.6.3.1. Primera iteración

Se mostrará el campo para realizar el registro correspondiente a cada usuario. Solo se pide los datos más importantes, para luego interactuar en el sistema.

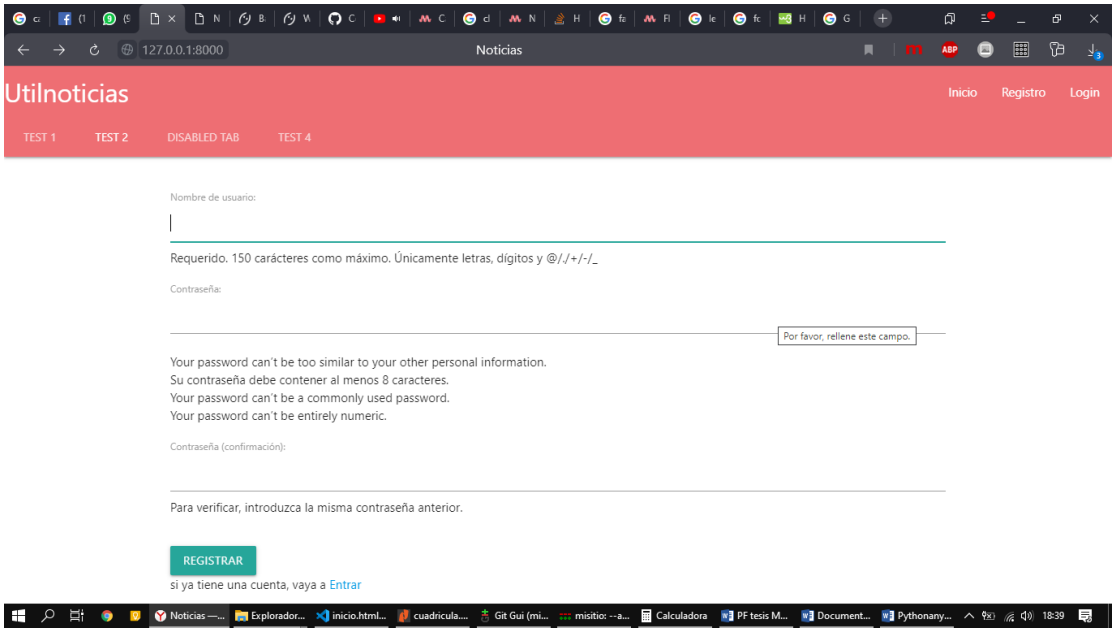


Figura 7 Registro de usuarios

Fuente: Elaboracion propia

4.6.3.2. Segunda iteración

Se mostrarán contenidos sin categorizar de las fuentes de noticias que han sido previamente establecidas, los resultados son presentados para todos sin discriminación ni clasificación. Se comprueba que es posible obtener los contenidos de las páginas web, aunque algunas no siempre están disponibles, lo cual requiere manejo de errores previo para que no interfieran con la carga de datos. Cuando una fuente no se muestra disponible, se excluye del proceso hasta la siguiente carga de información desde los RSS.

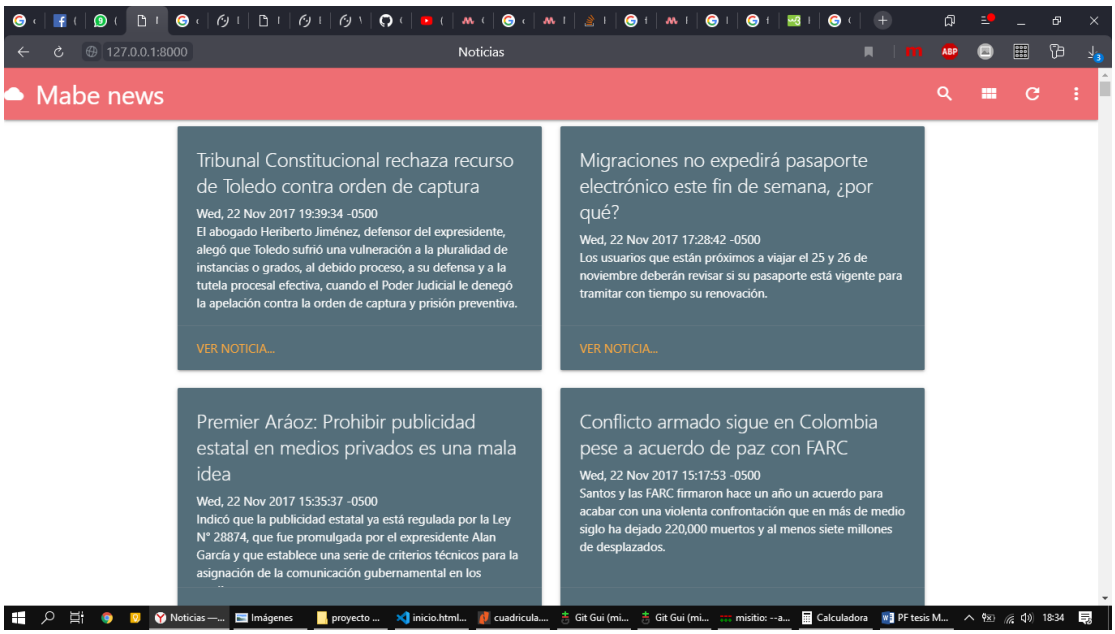


Figura 8 Mostrar noticias sin categorias

Fuente: Elaboración propia

4.6.3.3. Tercera interacción

Se muestran contenidos categorizados y se permite el registro de usuarios nuevos los cuales pueden ver noticias presentadas al azar desde las fuentes de datos predefinidas.

Figura 9 Noticias mostradas en categorías

Fuente: Elaboración propia

4.6.3.4. Cuarta iteración

Se adapta a dispositivos móviles mediante el diseño responsivo, considerando que el medio principal de acceso a internet es el móvil fue necesario hacer un diseño responsivo que se adapte a dispositivos con pantallas pequeñas. Para conseguir esto se hizo uso del framework "Materialize" junto con jQuery.

Figura 10 Diseño responsivo de la aplicación

Fuente:8 Elaboración propia

4.6.3.5. Quinta iteración

Se implementó el registro de usuarios nuevos y la recuperación de contraseña, se permiten los registros de preferencias de los usuarios y se calcula las recomendaciones para cada uno. Las categorías también se presentan y se han acomodado de forma que se puedan desplegar sin alterar el diseño del sitio.

Figura 11 Interface para móvil terminada de la aplicación

Fuente:9 Elaboración propia

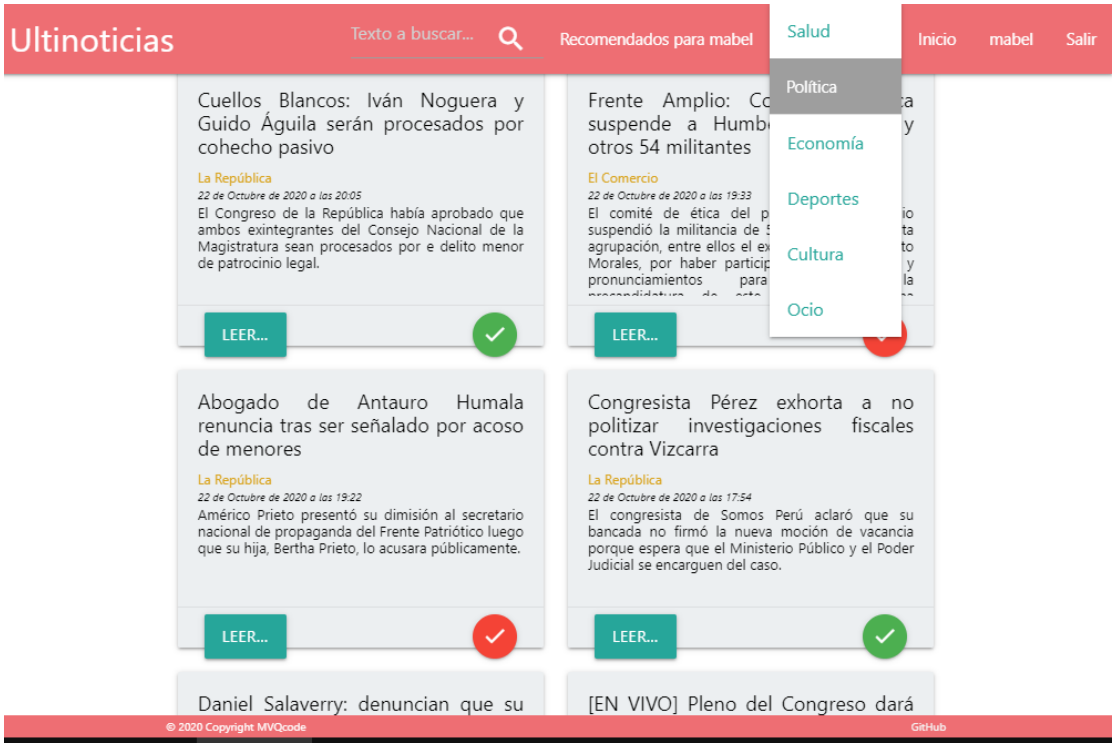


Figura 12 Interface para navegador de la aplicación

Fuente: Elaboración propia

Figura 13 Funcionalidad de registro de preferencias

Fuente: Elaboración propia

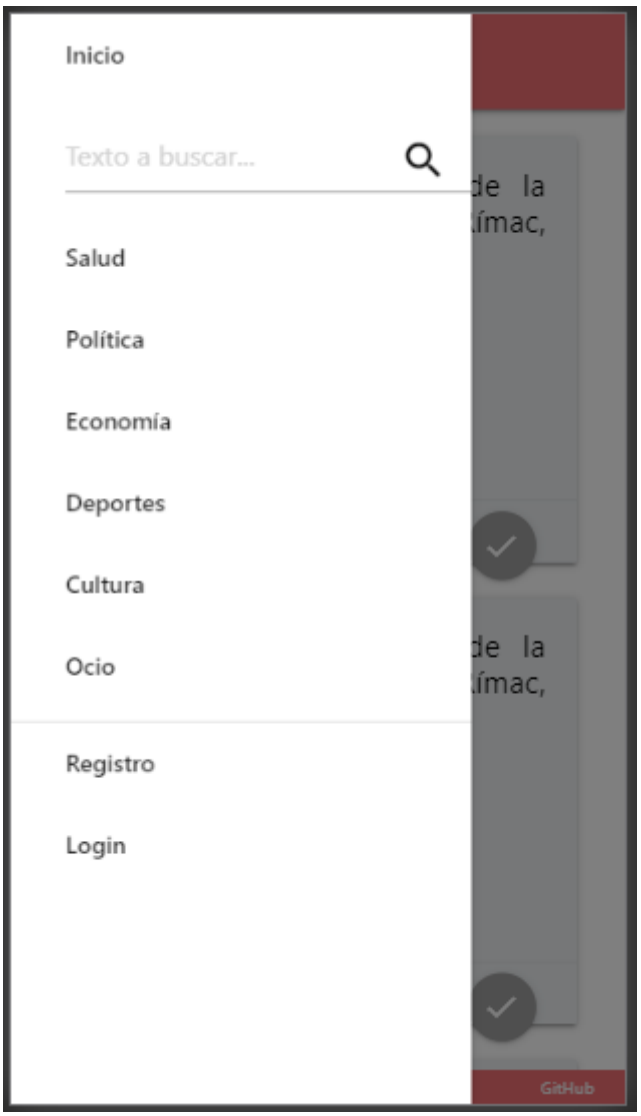


Figura 14 Menú de opciones en interface móvil

Fuente: 10Elaboración propia

4.6.4. Fase de producción

En esta fase se pone en funcionamiento de la aplicación para los usuarios. El código fuente se actualiza cada vez que hay una nueva característica de la fase previa.

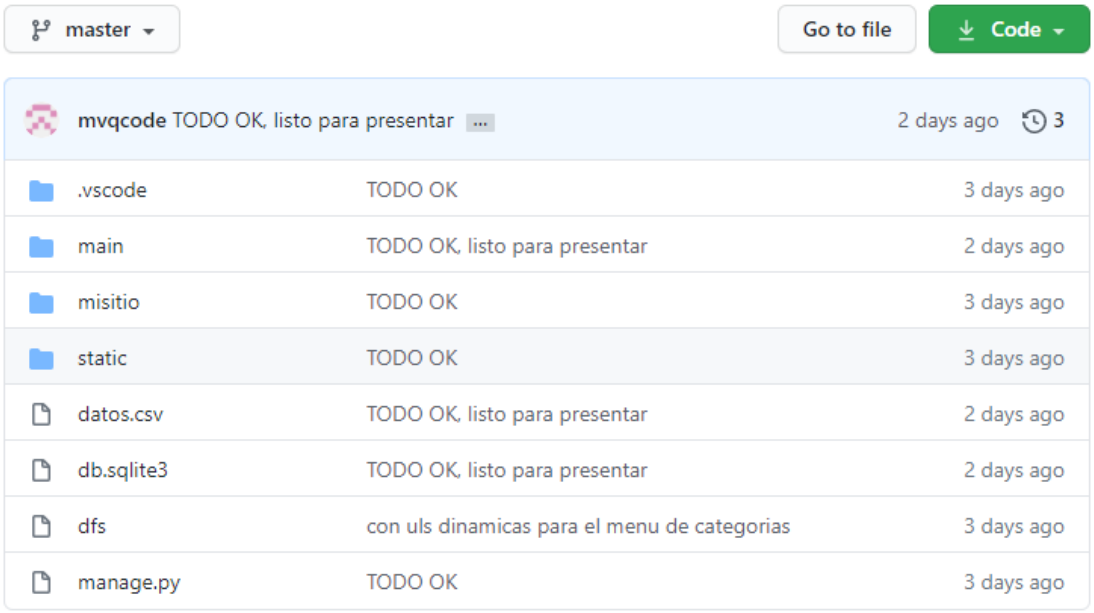


Figura 15 Repositorio en GitHub

Fuente: Elaboración propia

La URL de la aplicación es http://ultinoticias.pythonanywere.com

4.6.5. Fase de mantenimiento

En esta fase es en la que se puede agregar nueva funcionalidad, además de mantener el sistema corriendo, se pueden incorporar nuevas características.

En este caso en particular se limitará a mantener el sistema corriendo, dando el mantenimiento necesario y vigilando el correcto funcionamiento de la aplicación web/móvil.

Capítulo 5 RESULTADOS

5.1. Funcionalidad

Se mide a través de su complejidad, la funcionalidad no se puede medir directamente. Por lo que se establecen puntos de función (Pressman, 2002).

Parámetros de medida	Cuenta	Factor de peso			Subtotal
		Simple	Medio	Complejo	
Número de entradas de usuario	6	3	4	6	24
Número de salidas de usuario	3	4	5	7	15
Número de peticiones de usuario	5	3	4	6	20
Número de archivos	1	7	10	15	10
Número de interfaces	5	5	7	10	35

Tabla 18 Cálculo de puntos de función

Fuente: Elaboracion propia

Los valores de la variable Fi, se obtienen de los resultados que aprecian en la tabla 4.3, bajo la siguiente ponderación:

Escala	Complejidad
0	Sin influencia
1	Incidental
2	Moderado
3	Medio
4	Significativo
5	Esencial

Tabla 19 Valores de ajuste de complejidad

Fuente: Elaboracion propia

Pregunta	Ponderación
¿Requiere el software copias de seguridad y recuperación fiables?	5
¿Se requiere comunicación de datos?	5
¿Existen funciones de procesamiento distribuido?	3
¿Es crítico el rendimiento?	3
¿se ejecuta el sistema en un entorno operativo existente y fuertemente utilizado?	5
¿Requiere el software la entrada de datos interactiva?	5
¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	3
¿Se actualizan los archivos maestros de formas interactivas?	5
¿Son complejos las entradas, las salidas, los archivos o las peticiones?	3
¿Es complejo el procesamiento interno?	4
¿Se ha diseñado el código para ser reutilizable?	4
¿Están incluidas en el diseño la conversión y la instalación?	3
¿Se ha diseñado el software para soportar múltiples instalaciones en diferentes organizaciones?	4
¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	5
P(Fi) Total	57

Tabla 20 Ajuste de complejidad de punto de función

Fuente: Elaboración propia

Empleando la fórmula para hallar el PF y PF (máximo)

PF = 104*[0.65 + 0.01 * 57]

PF = 126.88

PF(Máximo) = 104* [0.65 + 0.01 * 70]

PF(Máximo) = 140.4

Con los valores máximos de ajuste de complejidad de Punto de Función, se tiene el siguiente resultado de Funcionalidad Real:

FUNCIONALIDAD = (126.88 / 140.4) * 100

FUNCIONALIDAD = 90.37%

Por lo tanto, la funcionalidad del sistema es representada en un 90.37% teniendo en cuenta el punto función máximo. Con esto se concluye que el sistema cumple con los requisitos funcionales de forma satisfactoria.

5.2. Usabilidad

Se espera que el software sea de fácil entendimientos y aprendizaje. Cabe resaltar que para la norma ISO 9126, la usabilidad no es afectada por la funcionalidad y eficiencia. La usabilidad está definida por los usuarios finales o asociados al software. (Pressman, 2002).

Para la medición de la usabilidad se tiene el siguiente cuestionario con los ajustes necesarios que se pueden observar en la siguiente tabla:

Escala	Descripción
1	Pésimo
2	Malo
3	Regular
4	Bueno
5	Muy Bueno

Tabla 21 Escala de ajustes de usabilidad
Fuente: Elaboración propia

Factor	Ponderación
¿Se ha satisfecho todos los requisitos establecidos por el software?	5
¿Es sencillo acceder a los datos?	5
¿Presenta suficiente ayuda durante el tiempo que accede al software?	4
¿Los informes son suficientemente representativos?	3
¿El software tiene la seguridad necesaria?	5
¿Está de acuerdo con la funcionalidad del software?	5
¿El software facilitará el trabajo que realizo?	5
Total	32

Tabla 22 Evaluación de usabilidad

Fuente: Elaboración propia

La usabilidad se calcula mediante la fórmula:

$$USABILIDAD = \frac{\left[\frac{\sum valor}{n} \times 100\right]}{5}$$

$$USABILIDAD = \frac{\left[\frac{32}{n7} \times 100\right]}{5}$$

$$USABILIDAD = 91.43\%$$

Por lo que la usabilidad de la aplicación es de 91.43% que interpretamos como la facilidad del usuario al interactuar con las interfaces del sistema.

5.3. Mantenibilidad

La mantenibilidad es la capacidad del software a ser modificado a nivel funcional, correcciones de mejoras y cambios en el entorno de software. (Pressman, 2002).

La mantenibilidad se determina mediante:

Mt = Número de módulos en la versión actual

Fi = Número de módulos en la versión actual que se han modificado

Fa = Número de módulos en la versión actual que se han añadido

Fd = Número de módulos en la versión anterior que se han añadido en la versión actual.

Así la mantenibilidad se calcula utilizando la siguiente relación:

$$MANTENIBILIDAD = [Mt - (Fa + Fi + Fd)] / Mt$$

Los valores para el sistema son los siguientes:

Mt = **7**

Fi = **1**

Fa = **0**

Fd = **0**

$$MANTENIBILIDAD = [7 - (0 + 1 + 0)] / 7$$

$$MANTENIBILIDAD = 0{,}86$$

La mantenibilidad es de 86% por lo que el producto de software contempla una cantidad de esfuerzo mínimo para cualquier cambio o modificación que requiera el software.

Por lo tanto, existe una probabilidad de un 86% de que el desarrollador pueda realizar cambios con facilidad en el código o estructura interna de software propuesto.

5.4. Portabilidad

La portabilidad es la capacidad del sistema para ser trasladado de un sistema a otro. (Pressman, 2002). La misma se puede calcular de la siguiente manera:

$$PORTABILIDAD = 1 - \left(\frac{\#\ de\ días\ para\ portar\ el\ sistema}{\#\ de\ días\ para\ implementar\ el\ sistema}\right)$$

$$\boldsymbol{PORTABILIDAD = 1 - \left(\frac{1}{60}\right)}$$

Por lo tanto, la portabilidad del sistema es de 98%, que significa que el sistema es fácil de transportar, es decir de ser llevado de un lugar a otro sin esfuerzo.

5.5. Resultado de la calidad del software

Para ponderar la calidad general del software, sacaremos la media de todas las medidas expresadas en porcentajes.

Características	Resultados %
Funcionalidad	90.37
Usabilidad	91.43
Mantenibilidad	86
Portabilidad	98
Evaluación de la Calidad Final	91.45

Tabla 23 Resultado general de la calidad del software

Fuente: Elaboración propia

La calidad del software es de 91.45%, lo que se interpreta como la satisfacción que tiene el usuario al interactuar con el software.

Capítulo 6 DISCUSIÓN

Previo a la implementación de la solución propuesta fueron definidos un total de 8 requisitos funcionales del sistema y 5 requisitos no funcionales. Todos estos requisitos fueron extraídos partiendo de los resultados del estudio previo del estado actual de las soluciones similares existentes y a entrevistas realizadas a los periodistas de los medios locales. La herramienta obtenida cuenta con 3 funcionalidades principales: el módulo de recopilación de información, el módulo de actualización de la información ya extraída y el módulo de recomendación.

Debido a que las aplicaciones descritas en los antecedentes no se encuentran disponibles y en línea, se realizó la comparación con dos productos del mercado de búsqueda de noticias:

- Comparación de la aplicación desarrollada con el agregador de noticias o lector de RSS "Feedly"
- Comparación de la aplicación desarrollada con la aplicación de "Google noticias"
- Prueba de aplicación con los periodistas de la radio "La Primera"

Comparación de la aplicación desarrollada con el agregador de noticias o lector de RSS "Feedly"

Se procedió a utilizar la aplicación "Feedly", para comparar las funcionalidades que ofrecía esta aplicación en comparación con la aplicación desarrollada. Se hizo la configuración de las fuentes de noticias a ser cargadas en la aplicación.

Comparación de la aplicación desarrollada con Google noticias

Se procedió a configurar las preferencias de Google noticias

Prueba de aplicación con los periodistas de la radio "La Primera"

Se seleccionó a 5 personas, periodistas y presentadores de la radio, para que prueben la aplicación "Ultinoticias", para que utilicen la aplicación en su Smartphone y en el navegador de la PC.

Para medir la usabilidad del producto se empleó un cuestionario de satisfacción. Este cuestionario también ayuda a medir que tan eficientemente realizan su labor los usuarios, y que tanto se pueden fidelizar al uso del producto. Este instrumento es cualitativo y por tanto subjetivo.

Las preguntas del cuestionario fueron:

P1: ¿La aplicación Ultinoticias es útil para usted?

P2: ¿La forma de navegación de la aplicación es fácil de comprender?

P3: ¿Qué le parece la presentación de las noticias?

P4: ¿Qué opina sobre la funcionalidad de la selección de noticias?

P5: ¿Cómo es la utilización de la aplicación en su móvil?

A las cuales puede responder en una escala del 1 al 5, siendo 1: muy malo y 5: excelente

Interpretación de los datos

Característica	Feedly	Google Noticias	Ultinoticias
Visualización de noticias nuevas	SI	SI	SI
Recomendaciones por temas de interés	NO	SI	SI
Elección de noticias favoritas	SI	SI	SI
Notificaciones	SI	SI	NO
Presentación de noticias locales	SI	SI	SI

Tabla 24 Comparación de aplicaciones

Fuente:11 Elaboración propia

Prueba de la aplicación en personas

Los resultados del cuestionario se muestran en el Anexo A, como se puede observar se obtuvo una calificación entre 4 y 5 por lo que se puede interpretar que es del agrado de los usuarios.

	P1	P2	P3	P4	P5
A	4	5	4	3	4
B	4	5	5	4	5
C	5	4	4	5	3
D	4	4	5	4	5

Tabla 25 Resumen de los resultados de la encuesta

Fuente: 12Elaboración propia

CONCLUSIONES

Se consiguió desarrollar el modelo de búsqueda basado en contenido, gracias al contenido publicado en las fuentes de noticias en formato RSS. Lo que permite hacer web Scraping de forma estructurada de modo que las noticias obtenidas tienen los mismos campos de datos y se pueden combinar para su procesamiento y visualización. Esto fue facilitado por el Framework Django para Python, debido a que Python implementa bibliotecas que facilitan la extracción de datos de los sitios web.

La metodología XP (Programación Extrema), permitió organizar el desarrollo del sistema web de recomendaciones de noticias, la aplicación web está desarrollada en un servidor web Django basado en Python con arquitectura MVT (modelo, vista, plantilla).

El desarrollo del algoritmo de recomendaciones basado en contenido, se apoyó en el algoritmo TF-IDF (Term frequency – Inverse document frequency) que permitió identificar palabras relevantes asignándole un valor alto y la cantidad de veces que aparece la palabra en la página web de noticias

RECOMENDACIONES

La aplicación desarrollada trabaja exclusivamente con fuentes de noticias que publiquen su contenido en formato RSS. No todas las páginas de noticias utilizan este formato. Para poder incluir más fuentes de noticias se recomienda complementar la carga de información con técnicas de Scraping basadas en HTTP.

Las recomendaciones basadas en contenido, son eficientes con pocos usuarios que generan mucha información de interacción con las noticias. En cuanto la aplicación se haga más popular, se puede complementar con técnicas de recomendación basadas en usuarios y semejanzas de sus perfiles.

REFERENCIAS BIBLIOGRÁFICAS

Aldo Capello, A. (2018). *Sistema de recomendaci ´on para textos legales*.

Barranco, M. J., Pérez, L. G., & Martínez, L. (2006). *Un Sistema de Recomendación Basado en Conocimiento con Información Lingüística Multigranular*. 645–664.

Big Data, I. A. (2018, December 27). *Machine Learning e Inteligencia Artificial: cómo funciona un sistema de recomendación*. https://itelligent.es/es/machine-learning-funciona-sistema-recomendacion/

Burke, R. (2002). *Hybrid Recommender Systems : Survey and*.

Chávez Arce, J. A. (2016). *FACULTAD DE CIENCIAS PURAS Y NATURALES NATURAL EN LA WEB "*.

D.Sergio, C. T. (2012). *Departamento de Ciencias de la Computación e Inteligencia Artificial Doctor en Informática*.

E.Peis;J.M.Morales del Castillo;A.Desgado López. (2008). *Sistema de Recomendacion*.

Glez Peña, Lourenco, López Fernández, R. J., & Fdez Riverola. (2013). *Web scraping technologies in an API world*. *15*(5), 788–797. https://doi.org/10.1093/bib/bbt026

Hernández, Vázquez Gómez, Berdejo Rincón, M. G., & Calderón Maldonado, I. O. (2015). *Metodologías para análisis político utilizando Web Scraping*. *95*, 113–121.

Huaman, J., & Quispe, M. (2019). *Modelo de búsqueda de productos alimenticios en supermercados online categoría abarrotes utilizando asistente virtual de tipo Chatbot y extracción de datos con Web Scraping*. 261. http://repositorio.utp.edu.pe/handle/UTP/2381

Ibáñez Micó, J. (2015). *Proyecto final de carrera Extracción de datos web usando técnicas de screen-scraping* . 1–55.

Letelier, P., & Penadés, C. (2012). *Métodologías ágiles para el desarrollo de software : eXtreme Programming (XP)*.

MinjusDH. (2020, April 8). *Ministerio de Justicia y Derechos Humanos en Twitter: "Quienes desinformen a la ciudadanía con noticias falsas para obtener un beneficio o perturbar la tranquilidad pública serán sancionados con pena privativa de la libertad. Comparte solo información oficial del #COVID19, ¡no difundas #FakeNews! #PerúEstáEnNuestrasManos https://t.co/S0mfQ1UITI" / Twitter*. https://twitter.com/MinjusDH_Peru/status/1247871817815150592?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1247871817815150592%7Ctwgr%5E&ref_url=https%3A%2F%2Flarepublica.pe%2Fsociedad%2F2020%2F04%2F08%2Fcoronavirus-en-peru-carcel-hasta-6-anos-para-quienes-difundan-informacion-falsa-ministerio-de-justicia-atmp%2F

Ml Y Bayes.Pdf. (n.d.).

Orts, J. (2020). *Tema 5: La noticia*. http://www.periodismo-online.de/la-prensa/prensa/tema-5-la-noticia/

Pazzani, M. J., & Billsus, D. (2007). *Content-Based Recommendation Systems*. 325–341.

Proyecto, M. D. E. L. (2015). *MadridManía : Un sistema de recomendación social y de ocio para grupos*.

RAE-La 23a Edición. (2019a). *periodismo | Definición | Diccionario de la lengua española | RAE - ASALE*. https://dle.rae.es/periodismo

RAE-La 23a Edición. (2019b). *periodista | Definición | Diccionario de la lengua española | RAE - ASALE*. https://dle.rae.es/periodista

Un panel de expertos analizó la importancia de valorar la producción de las noticias, las fake news y el futuro del periodismo - Infobae. (n.d.). Retrieved August 16, 2020, from https://www.infobae.com/sociedad/2020/08/04/un-panel-de-expertos-analizo-la-importancia-de-valorar-la-produccion-de-las-noticias-las-fake-news-y-el-futuro-del-periodismo/

Universidad nacional de piura. (2019).

Vargiu, E., & Urru, M. (2012). Exploiting web scraping in a collaborative

filtering- based approach to web advertising. *Artificial Intelligence Research*, *2*(1). https://doi.org/10.5430/air.v2n1p44

Funes, A., & Dasso, A. (2014). *Evaluación de Frameworks para Aplicaciones Web Resumen Contexto Introducción*. 555–559.

Jeff, F., & Paul Bissex. (2008). *Python Web Development with Django*. 24 de Octubre. https://books.google.es/books?hl=es&lr=&id=M2D5nnYlmZoC&oi=fnd&pg=PT31&dq=django+python&ots=vZSLEraQOX&sig=Rt3dfhWdikjkL3UpYjN_5A1QDcA#v=onepage&q=django python&f=false

Maarif, A. A. (2015). Penerapan Algoritma TF-IDF untuk Pencarian Karya Ilmiah. *Dokumen Karya Ilmiah | Tugas Akhir | Program Studi Teknik Informatika - S1 | Fakultas Ilmu Komputer | Universitas Dian Nuswantoro Semarang*, *5*, 4. mahasiswa.dinus.ac.id/docs/skripsi/jurnal/15309.pdf

Martínez Acosta, D. de J. (2014). *Herramienta para la generación automática del código fuente para aplicaciones con arquitectura modelo vista controlador (MVC) bajo desarrollo dirigido por modelos textuales (MDD). Universidad Nacional de Colombia*, 87. http://www.bdigital.unal.edu.co/45826/

Montero Molina, B., Cevallos Vite, H., & Dávila Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales Revista Multidisciplinaria de Investigación ISSN: 2550-6862*, *2*(17), 114–121. https://www.researchgate.net/publication/327537074_Metodologias_agiles_frente_a_las_tradicionales_en_el_proceso_de_desarrollo_de_software

Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico.*

ANEXOS

Estructura en XML de las fuentes RSS

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<rss xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:media="http://search.yahoo.com/mrss/"
 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:mi="http://schemas.ingestion.microsoft.com/common/"
 xmlns:dcterms="http://purl.org/dc/terms/" xmlns:content="http://purl.org/rss/1.0/modules/content/"
 version="2.0" xml:base="http://peru.com">
  ▼<channel>
     <title>Peru.com</title>
     <link>https://peru.com/actualidad/politicas</link>
    ▼<description>
       <![CDATA[ Noticias de Políticas | Actualidad | Noticias del Perú, Martín Vizcarra, Congreso de
       la República, Perú, Lima, Mundial Rusia 2018, Feriados 2018 | Peru.com ]]>
     </description>
     <language>es</language>
     <copyright>Peru.com</copyright>
     <lastBuildDate>Tue, 03 Nov 2020 19:12:28 -0500</lastBuildDate>
     <ttl>15</ttl>
    ▼<image>
       <title>Peru.com</title>
       <url>http://cdn.perucom3.e3.pe/f/i/logo_perucom.png</url>
       <link>https://peru.com</link>
     </image>
    ▼<item>
      ▼<title>
         <![CDATA[ Realizarán campaña de vacunación contra difteria, neumococo e influenza ]]>
       </title>
      ▼<description>
         <![CDATA[ <p>Iniciará el miércoles 4 de noviembre a las 9 a.m. en la losa deportiva de la
         Residencial Orbegoso, ubicada frente al Mercado Pando. Esta será para vacunar contra la
         influenza y el neumococo a adultos mayores.</p> ]]>
       </description>
       <pubDate> Tue, 03 Nov 2020 18:15:55 -0500 </pubDate>
       <dcterms:modified>2020-11-03T18:15:55-05:00</dcterms:modified>
       <dcterms:alternative> Realizarán campaña de vacunación contra difteria, neumococo e influenza
       </dcterms:alternative>
       <link> https://peru.com/actualidad/mi-ciudad/realizaran-campana-vacunacion-contra-difteria-
       neumococo-influenza-noticia-615026 </link>
       <guid isPermaLink="false"> https://peru.com/actualidad/mi-ciudad/realizaran-campana-
       vacunacion-contra-difteria-neumococo-influenza-noticia-615026 </guid>
       <dc:type>Artículo</dc:type>
```

```python
from django.db import models  # hereda del modelo base de django
from datetime import datetime
from django.contrib.auth.models import User
from django.db.models.signals import post_save
from django.dispatch import receiver
# Create your models here.


class Perfil(models.Model):
    usuario = models.OneToOneField(User, on_delete=models.CASCADE)
    organizacion = models.TextField(max_length=30, default="Ninguna")
    email = models.EmailField(blank=False)

    def __str__(self):  # version imprimible
        return self.usuario.username


def crear_usuario_perfil(sender, instance, created, **kwargs):
    if created:
        Perfil.objects.create(usuario=instance)


def guardar_usuario_perfil(sender, instance, **kwargs):
    instance.pefil.save()


class Categoria(models.Model):
    categoria = models.CharField(max_length=20, primary_key=True)
    descripcion = models.TextField()


class Fuente(models.Model):
    diario = models.CharField(max_length=50)
    url = models.TextField(max_length=250)
    categoria = models.ForeignKey(
        Categoria, on_delete=models.CASCADE, null=True)
    activo = models.BooleanField(default=False)


class Noticia(models.Model):
    id_noticia = models.IntegerField(primary_key=True)
    titulo = models.TextField()
    descripcion = models.TextField()
    url = models.TextField()
    urlImagen = models.TextField()
    fecha = models.DateTimeField()
```

```
    autor = models.CharField(max_length=50)
    nombre = models.CharField(max_length=50)
    fuente = models.ForeignKey(Fuente, on_delete=models.CASCADE, null=T
rue)


class Preferencia(models.Model):
    id_noticia = models.ForeignKey(
        Noticia, on_delete=models.CASCADE, null=True)
    usuario = models.CharField(max_length=50)
    preferencia = models.BooleanField(default=False)

    def __str__(self):
        return str(self.id_noticia)
```

Vistas

```
from time import mktime
from datetime import datetime
import unicodedata
from sklearn.metrics.pairwise import linear_kernel
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
# import requests
import pprint
import random
import feedparser
from django.shortcuts import render, redirect
from django.http import HttpResponse
from django.contrib.auth.forms import UserCreationForm, AuthenticationF
orm, forms
from django.contrib.auth.models import User
from django.contrib.auth.views import LoginView
from django.views.generic import CreateView, TemplateView
from django.contrib.auth import login, logout, authenticate
from django.contrib import messages
from django.utils import timezone
# from django.core.urlresolvers import reverse_lazy
import hashlib

from .models import Preferencia, Noticia, Perfil, Fuente, Categoria

from .forms import SignUpForm


class articulo:
    '''
    Clase que contiene temporalmente cada artículo de noticias
    '''

    def __init__(self, article):
        h = hashlib.md5()
        h.update(article['title'].encode('utf-8'))
        self.id = int(h.hexdigest(), 16) % 10000000
        self.titulo = article['title']
        self.descripcion = article['description']
        self.url = article['url']
        self.urlImagen = article['urlToImage']
        self.fecha = article['publishedAt']
        self.autor = article['author']
        self.nombre = article['source']['name']


def extraerArticulos(feeds):
```

```
    '''
    Recibe una lista de feeds cada item de la lista tiene los atributos
 'diario' y 'url'
    Devuelve un iterable conteniendo los objetos articulo obtenidos del
 feed
    '''
    for feed in feeds:
        # Lee el RSS contenido en el feed y obteien un diccionario dond
e cada entrada es un dato del artículo
        rss = feedparser.parse(feed['url'])
        for item in rss['entries']:
            article = dict()
            if 'title' in item:
                article['title'] = item['title']
                article['description'] = item['summary']
                article['url'] = item['link']
                article['urlToImage'] = '#'
                article['publishedAt'] = timezone.make_aware(datetime.s
trptime(
                    item['published'][5:25], '%d %b %Y %H:%M:%S'), time
zone.get_default_timezone())  # convierte la hora en formato utc
                article['author'] = ''
                article['source'] = dict()
                article['source']['name'] = feed['diario']
                yield articulo(article)


def obtenerFuentes(categorias):
    articulos = []  # aqui se acumularan los objetos articulo
    # obtiene todas las fuentes de la base de datos
    # aqui se guardaran las fuentes en un diccionario por categorias
    todasFuentesCategoria = {}
    for categoria in categorias:
        # seleccion por un Foreign Key
        fuentes = Fuente.objects.filter(
            categoria__categoria=categoria['categoria']).filter(activo=
True).values()
        articulosCategoria = list(extraerArticulos(fuentes))
        articulosCategoria.sort(key=lambda x: x.fecha, reverse=True)
        articulos = articulos + articulosCategoria
        todasFuentesCategoria.setdefault(
            categoria['categoria'], articulosCategoria)

    return articulos, todasFuentesCategoria


# devuelve el objeto artículo de una lista de artículo
```

```
def grabaNoticia(articulo):
    loquehay = Noticia.objects.filter(id_noticia=articulo.id)
    if len(loquehay) == 0:
        instancia = Noticia.objects.create(
            id_noticia=articulo.id,
            titulo=articulo.titulo,
            descripcion=articulo.descripcion,
            url=articulo.url,
            urlImagen=articulo.urlImagen,
            fecha=articulo.fecha,
            autor=articulo.autor,
            nombre=articulo.nombre,
        )
    else:
        instancia = loquehay.last()
    instancia.save()
    return instancia


def normalizatexto(texto):
    '''
    Normaliza el texto para evitar que caracteres extraños entren al co
nteo de palabras
    '''
    return unicodedata.normalize('NFKD', texto).encode('ASCII', 'ignore
').strip().lower()


def calculaDistancias(articulos):

    articulos.sort(key=lambda x: x.fecha, reverse=True)

    f = open("datos.csv", "w", encoding="utf-8")
    f.write('id,description\n')
    for articulo in articulos:
        linea = "" + str(articulo.id) + ",\"" + articulo.titulo + \
            "-" + articulo.descripcion + "\"\n"
        f.write(linea)
    f.close()

    # Prepara el vector de puntajes para hacer las recomendaciones

    ds = pd.read_csv("datos.csv", usecols=[
                     "id", "description"], encoding="latin")

    tf = TfidfVectorizer(analyzer='word', ngram_range=(1, 3),
                         min_df=0, stop_words='english')
    tfidf_matrix = tf.fit_transform(ds['description'])
```

```
    cosine_similarities = linear_kernel(tfidf_matrix, tfidf_matrix)

    results = {}

    for idx, row in ds.iterrows():
        similar_indices = cosine_similarities[idx].argsort()[:-100:-1]
        similar_items = [(cosine_similarities[idx][i], ds['id'][i])
                         for i in similar_indices]

        results[row['id']] = similar_items[1:]

    return results


categorias = Categoria.objects.all().values()
articulos, todosXcategoria = obtenerFuentes(categorias)
results = calculaDistancias(articulos)


# def item(id):
#     return ds.loc[ds['id'] == id]['description'].tolist()[0].split('
- ')[0]
def refrescar(request):
    categorias = Categoria.objects.all().values()
    articulos, todosXcategoria = obtenerFuentes(categorias)
    results = calculaDistancias(articulos)
    return redirect("main:homepage")


def recommend(item_id, num):
    recs = results[item_id][:num]
    return recs


class SignUpView(CreateView):
    '''
    Esto crea la vista que muestra el formulario sobrecargado que acept
a el email en base al modelo Perfil
    '''
    model = Perfil
    form_class = SignUpForm

    # sobrecargamos la función que valida el formulario para que si se
ha ingresado los datos correctos se permita el longin

    def form_valid(self, form):
        if form.is_valid():
            usuario = form.save()
            nombre_usuario = form.cleaned_data.get(
```

```
                'username')  # obtiene el nombre del usuario
            messages.success(
                self.request, f"Nueva Cuenta Creada : {nombre_usuario}"
)
            login(self.request, usuario)
            messages.info(
                self.request, f"Has sido logueado como {nombre_usuario}
")
            return redirect("main:homepage")
        else:
            for msg in form.error_messages:
                messages.error(
                    self.request, f"{msg}: {form.error_messages[msg]}")
        return render(self.request, "main/perfil_form.html", {"form": f
orm})

    def form_invalid(self, form):
        for msg in form.error_messages:
            messages.error(
                self.request, f"{form.error_messages[msg]}")
        return render(self.request, "main/perfil_form.html", {"form": f
orm})


def perfiluser(request):
    usuario_actual = request.user
    consultaPreferencias = Preferencia.objects.filter(
        usuario=usuario_actual)
    return render(request, "main/bienvenida.html", {"news": consultaPre
ferencias, })


def recuperacion(request):
    return render(request, "main/recuperacion.html")


def usuariorec(request):
    return render(request, "main/usuariorec.html")


def categoria(request, nomcat=None):
    usuario_actual = request.user
    consultaPreferencias = Preferencia.objects.filter(
        usuario=usuario_actual)
    preferencias = [
        preferencia.id_noticia.id_noticia for preferencia in consultaPr
eferencias]
```

```
    return render(request, "main/inicio.html", {"news": todosXcategoria
[nomcat], "categoria": nomcat, "activa": nomcat, "categorias": categori
as, "preferencias": preferencias, })


def politica(request):
    # random.shuffle(articulosPolitica)
    usuario_actual = request.user
    consultaPreferencias = Preferencia.objects.filter(
        usuario=usuario_actual)
    preferencias = [
        preferencia.id_noticia.id_noticia for preferencia in consultaPr
eferencias]
    return render(request, "main/inicio.html", {"news": articulosPoliti
ca, "categoria": "politica", "activa": "politica", "categorias": catego
rias, "preferencias": preferencias, })


def economia(request):
    # random.shuffle(articulosEconomia)
    usuario_actual = request.user
    consultaPreferencias = Preferencia.objects.filter(
        usuario=usuario_actual)
    preferencias = [
        preferencia.id_noticia.id_noticia for preferencia in consultaPr
eferencias]
    return render(request, "main/inicio.html", {"news": articulosEconom
ia, "categoria": "economia", "activa": "economia", "categorias": catego
rias, "preferencias": preferencias, })


def salud(request):
    # random.shuffle(articulosSalud)
    usuario_actual = request.user
    consultaPreferencias = Preferencia.objects.filter(
        usuario=usuario_actual)
    preferencias = [
        preferencia.id_noticia.id_noticia for preferencia in consultaPr
eferencias]
    return render(request, "main/inicio.html", {"news": articulosSalud,
 "categoria": "salud", "activa": "salud", "categorias": categorias, "pr
eferencias": preferencias, })


def busqueda(request):
    usuario_actual = request.user
    consultaPreferencias = Preferencia.objects.filter(
        usuario=usuario_actual)
    preferencias = [
```

```
        preferencia.id_noticia.id_noticia for preferencia in consultaPr
eferencias]
    textobuscado = normalizatexto(request.GET['search'])
    articulosBusqueda = [
        articulo for articulo in articulos if textobuscado in normaliza
texto(articulo.titulo)]
    return render(request, "main/inicio.html", {"news": articulosBusque
da, "categoria": "Busqueda", "activa": "Busqueda", "categorias": catego
rias, "preferencias": preferencias, })


def recomendados(request):
    usuario_actual = request.user
    consultaPreferencias = Preferencia.objects.filter(
        usuario=usuario_actual)
    preferencias = [
        preferencia.id_noticia.id_noticia for preferencia in consultaPr
eferencias]
    # print("Preferencias:", consultaPreferencias)
    if len(preferencias) == 0:  # si no hay preferencias, puede signifi
car que nunca eligió nada o que no se ha logueado
        idUltimaNoticia = None
    else:
        # si hay preferencias previas elegimos la última como referenci
a para la consulta
        idUltimaNoticia = preferencias[-1]
    # si no hubo ultima noticia y el id elegido no está en el computo d
e resultados
    if idUltimaNoticia == None or not idUltimaNoticia in results:
        elegido = random.randint(0, len(articulos))
        id_elegido = articulos[elegido].id
    else:
        id_elegido = idUltimaNoticia
    recomendaciones = recommend(item_id=id_elegido, num=20)
    ids_recomendados = [item[1] for item in recomendaciones]
    articulosrecomendados = [
        articulo for articulo in articulos if articulo.id in ids_recome
ndados and not articulo.id in preferencias]
    # print("PREFERENCIAS", preferencias)
    # print("ARTICULO", [art.descripcion for art in articulosrecomendad
os])
    return render(request, "main/inicio.html", {"news": articulosrecome
ndados, "categoria": "recomendados", "activa": "recomendados", "categor
ias": categorias, "preferencias": preferencias, })


def check(request):
    def recuperaNoticia(articulos, idn):
        for articulo in articulos:
```

```
            if articulo.id == int(idn):
                return articulo
    noticia = request.POST['id']
    if noticia == '':
        return HttpResponse('')
    # check = request.POST['click']
    print("ID:", noticia)
    usuario = request.POST['usuario']
    consulta = Preferencia.objects.filter(
        usuario=usuario).filter(id_noticia=noticia)
    if (len(consulta) == 0):
        nuevanoticia = recuperaNoticia(
            articulos, noticia)  # obtiene el objeto noticia
        print("NOTICIA:", nuevanoticia)
        noticiagrabar = grabaNoticia(nuevanoticia)
        nuevocheck = Preferencia.objects.create(
            id_noticia=noticiagrabar, preferencia=True, usuario=usuario
)
        nuevocheck.save()
        return HttpResponse('true')
    consulta.delete()
    return HttpResponse('false')


def registro(request):
    form = UserCreationForm()
    forms.CharField
    if request.method == "POST":
        form = UserCreationForm(request.POST)

        if form.is_valid():
            usuario = form.save()
            nombre_usuario = form.cleaned_data.get(
                'username')  # obtiene el nombre del usuario
            # crea un mensaje para el usuario
            messages.success(
                request, f"Nueva Cuenta Creada : {nombre_usuario}")
            login(request, usuario)
            messages.info(request, f"Has sido logueado como {nombre_usu
ario}")
            return redirect("main:homepage")
        else:
            for msg in form.error_messages:
                # print("ERROR:", msg)
                messages.error(request, f"{msg}: {form.error_messages[m
sg]}")

    return render(request, "main/registro.html", {"form": form})
```

```python
def logout_request(request):
    logout(request)
    messages.info(request, "Saliste exitosamente")
    return redirect("main:homepage")


def login_request(request):
    if request.method == "POST":
        form = AuthenticationForm(request, data=request.POST)
        if form.is_valid():
            usuario = form.cleaned_data.get('username')
            contraseña = form.cleaned_data.get('password')
            user = authenticate(username=usuario, password=contraseña)

            if user is not None:
                login(request, user)
                messages.info(request, f"Estas logueado como {usuario}"
)
                return redirect("main:homepage")
            else:
                messages.error(request, "Usuario o contraseña equivocad
a")
        else:
            messages.error(request, "Usuario o contraseña equivocada")

    form = AuthenticationForm()
    return render(request, "main/login.html", {"form": form})
```