



Facultad 6

Título: Desarrollo de un algoritmo para la personalización automática de la interfaz de la Plataforma VideoWeb a partir de las preferencias del usuario.

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

Autor: Lusmey Elejarde Díaz

Tutor: Ing. Ángel Dayán Marín Abreu

Ciudad de la Habana, 30 Junio, 2011

Año 53 de la Revolución



“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento, porque precisamente es lo que más estamos sembrando; lo que más estamos sembrando son oportunidades a la inteligencia (...)”

Fidel Castro



Declaración de Autoría

Yo: Lusmey Elejarde Díaz declaro que soy la única autora del trabajo titulado “Desarrollo de un algoritmo para la personalización automática de la interfaz de la Plataforma VideoWeb a partir de las preferencias del usuario”, y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Lusmey Elejarde Díaz

Ángel Dayán Marín Abreu

Firma del Autor

Firma del Tutor



Opinión del Tutor



Datos de Contacto

Tutor: Ing. Ángel Dayán Marín Abreu.

- ✓ Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, 2011.
- ✓ Analista principal del proyecto VideoWeb, perteneciente al centro GeySed de la facultad 6, Universidad de las Ciencias Informáticas.

Correo electrónico: admarin@uci.cu



Opiniones y Avaes



Dedicatoria

A las dos personas que me dieron la vida y adoro con todas las fuerzas de mi corazón, que me guiaron en todo momento, que me aman y protegen siempre: mis padres, que la vida no me permitió tenerlos en estos momentos a mi lado, pero sé que donde estén estarán orgullosos de mí.

A mi familia y todas aquellas personas que han compartido conmigo buenos y malos momentos.





Agradecimientos

A mi tesoro más grande: mi mamita, por su confianza, amor y dedicación, que sin su apoyo incondicional no hubiese logrado seguir adelante, sin su sacrificio y abnegación, sus sabios consejos, por compartir alegrías y tristezas, por la paciencia con la que ha sabido guiarme por este mundo.

A mi hermana, Lismary y mi cuñado Fide por demostrarme que sí podía lograr mis sueños.

A mi mima y mi pipo, mis viejitos lindos, que son lo más importante de mi vida, que soportan mis malcriadeces y nunca dejan de apoyarme.

Alejandro, mi Cosita, que soportó mis momentos de pesimismo e indicó cómo levantarme durante 5 años.

A mis tíos Lisy, Achy, Osmany y a mis abuelos paternos por estar siempre ahí cuando los necesite, a todos mis primos y demás familiares.

A Lien y Liennys por ser tan buenas conmigo.

A mi suegra por ser tan buena y preocuparse tanto por mí, gracias.

A Pedro, por su ayuda en todo momento, gracias.

A mi tutor, a la oponente y al tribunal de tesis por su apoyo, comprensión y entrega desde el primer día.

A mis compañeras de tesis: Nixys y Diana.

A mis viejos y nuevos amigos.

A mi niña, mi centauro asesino e inseparable compañera de cuarto: Elizabetha que compartió conmigo los buenos y malos momentos durante 5 años.

A mis hermanitas: Nane, Cecy, La Caly, Ody y Yudi por ser tan buenas amigas.

A mis hermanitos: A Pachico y al Rafa.

A mis Pellys: José y Orly.

A mis compañeras de todos estos años: Melo, Sachie, Yara, Ali, La flaca.

A Merlín que aunque solo compartimos este año es una excelente compañera.



A mis clarias del 202: la Mailen (la madre de los 4 hijos), A Grechita (Energía 21), a las Yanet (la clara y la bolla), a la Miche (la máquina asesina), a Leidys (la 7ma puerta), a Eliadys (la jefa del conjunto pinareño) y a la pegadas de Palmolive (Yanet) y Mimita.

A nuestro Comandante en Jefe Fidel Castro Ruz por hacer este sueño realidad.

A todas las personas que de una forma u otra colaboraron en el desarrollo de esta investigación y en mi preparación como futura Ingeniera en Ciencias Informáticas.

A todas aquellas personas que de una forma u otra han colaborado con el desarrollo de esta investigación.

Gracias por mantener la confianza de que iba a llegar a ser una gran profesional.

A todos, ¡muchas gracias!



Resumen

Actualmente la personalización es una práctica muy vigente, pues se utiliza en casi todas las aplicaciones donde se manipulan diferentes contenidos, con el objetivo de satisfacer los gustos y necesidades de los clientes. En la Universidad de las Ciencias Informáticas (UCI) el proyecto Plataforma VideoWeb del Centro de Geoinformática y Señales Digitales (GeySed) de la Facultad 6, encargado de la transmisión y publicación de contenidos multimedia, no cuenta con una interfaz capaz de satisfacer las necesidades de los usuarios; debido a que la interfaz y los contenidos de la plataforma se muestran de igual manera para todos los usuarios que interactúan con ella, provocando que los usuarios consuman mucho tiempo para encontrar sus contenidos preferidos.

Para materializar la idea anteriormente expresada fue llevada a cabo esta investigación, luego de un estudio detallado sobre las tendencias actuales del comportamiento de los usuarios en la web, se seleccionaron el Análisis de texto y el Contexto de usuario como técnicas de extracción de información para el análisis y procesamiento de los datos recogidos de la base de datos de la plataforma. El lenguaje de programación seleccionado para la implementación del algoritmo fue PHP por las ventajas que presenta este lenguaje en la construcción de sitios web, también se definieron la metodología de desarrollo y las herramientas a utilizar para la implementación del algoritmo y el prototipo funcional.

Como resultado de la investigación, se implementaron y probaron las funcionalidades que debería cumplir el algoritmo, se implementó un prototipo funcional para validar que el algoritmo cumpliera los objetivos trazados con la investigación realizada.

Palabras claves: Personalización, prototipo funcional, plataforma VideoWeb, interfaz, usuario.



Abstract

Currently, the personalization is a practice very much alive, as is used in almost all applications where different content is handled in order to satisfy the tastes and needs of customers. At the University of Informatics Sciences (UCI) VideoWeb Platform project of the Center for Geoinformatics and Digital Signals (GeySed), Faculty 6, responsible for the transmission and publication of multimedia content, it has an interface capable of meeting the needs of users, because the interface and the contents of the platform shown equally to all users who interact with it.

To materialize the idea expressed above was carried out this research after a detailed study of current trends in the behavior of web users were selected text analysis and user context as information extraction techniques to analysis and processing of data collected from the database platform. The programming language chosen for implementation of the algorithm was PHP the advantages of this language in building websites, also defined development methodology and tools used to implement the algorithm. As a result of the investigation, implement and test the features that should satisfy the algorithm, we implemented a functional prototype to validate the algorithm met the stated objectives to the research.

Keywords: Personalization, functional prototype, VideoWeb platform, interface, user.



Índice

Resumen	9
Abstract	10
Introducción	17
CAPÍTULO 1. “Fundamentación Teórica”	20
Introducción	20
1. Conceptos asociados al dominio del problema.	20
1.1 Principios de la personalización.	21
1.1.1 Variables para el proceso de personalización.	21
1.2 Descripción de la situación problemática.	22
1.3 Objeto de estudio.	23
1.3.1 Técnicas para solicitar la información al usuario.	23
1.3.1.1 Calificaciones explícitas (<i>Explicit ratings</i>).	23
1.3.1.2 Calificaciones implícitas (<i>Implicit ratings</i>).	23
1.3.2 Técnicas de extracción de información.	24
1.3.2.1 Reconocimiento de nombres de entidades (<i>Named entity recognition</i>).	24
1.3.2.2 Análisis de texto (<i>Text analytics</i>).	25
1.3.2.3 Minería de texto (<i>Text mining</i>).	25
Principales técnicas utilizadas en la minería de textos.	25
1.3.2.4 Contexto de usuario	26
Creación del contexto de usuario.	26
Perfil de usuario:	27



Conclusiones Parciales del Capítulo 1	28
CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”	29
Introducción	29
2.1 Arquitectura.	29
2.1.1 Arquitectura en 3 Capas.	29
2.1.2 Modelo Cliente/Servidor.....	30
2.1.3 Arquitectura Modelo Vista Controlador (MVC).....	31
2.2 Lenguaje de Programación.	33
Principales ventajas de PHP.....	33
2.2.2 C++.....	34
¿Por qué PHP?.....	35
2.3 Herramientas de Desarrollo.	35
2.4 Variables obtenidas del trabajo con la base de datos de la plataforma.....	36
2.5 Requisitos Funcionales	37
2.6 Requisitos no Funcionales	38
2.6.1 Restricciones en el diseño y la implementación.	38
2.7 Definición de los Actores del Sistema.	38
2.8 Diagrama de Caso de Uso del Sistema (DCUS).	39
2.9 Descripción de los Casos de Usos.....	40
2.10 Técnicas seleccionadas para el desarrollo del algoritmo.....	42
Conclusiones Parciales del Capítulo 2.....	44
CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”	45
Introducción	45



3.1 Desarrollo del algoritmo para la personalización de la interfaz de la plataforma VideoWeb.....	45
3.1.1 Listado de las tablas utilizadas para la implementación del algoritmo de personalización.....	45
Descripción de las tablas y atributos seleccionados para el desarrollo del algoritmo.	46
Tabla 6 Descripción de la tabla Campo_valor. Fuente: Elaboración Propia.	47
3.2 Implementación del algoritmo para personalizar la interfaz de la plataforma VideoWeb.	53
3.2.1 Funcionalidades realizadas para el desarrollo del algoritmo.	53
Funcionalidad getContentido.	53
Es la funcionalidad principal del algoritmo, encargada de retornar las recomendaciones con los contenidos que se encuentran dentro de las preferencias de los usuarios.	53
Funcionalidad getContentidoAux.	54
Funcionalidad getContentidoPersistente.....	57
Es la funcionalidad encargada de retornar los contenidos que cumplen con los campos que se encuentran dentro de las preferencias de los usuarios.	57
Funcionalidad inHistorial.....	58
Es la funcionalidad encargada de comprobar si un archivo multimedia se encuentra en el historial de un usuario para no mostrarle al mismo contenidos repetidos.....	58
Funcionalidad getContentidoAleatorio.	58
Es la funcionalidad encargada de mostrarles a los usuarios contenidos aleatorios en caso que no se encuentre ninguno que sea de su interés o que el mismo no cuente con un historial en la base de datos.	58
3.3 Validación del algoritmo de personalización a través de un prototipo funcional.	58
3.4 Resultados del algoritmo desarrollado para personalizar la interfaz de la plataforma VideoWeb.	62
Conclusiones del capítulo 3.....	63



Conclusiones Generales.....	64
Recomendaciones	65
Referencias Citadas.....	66
Referencias Bibliografías	68
Glosario de Términos.....	69



Índice de Tablas

Tabla 1 Variables obtenidas de la base de datos. Fuente: Elaboración propia.	36
Tabla 2 Descripción de los actores del sistema. Fuente: Elaboración propia.	39
Tabla 3 Personalizar interfaz de usuario. Fuente: Elaboración propia.	40
Tabla 4 Notificar nuevo contenido. Fuente: Elaboración propia.	41
Tabla 5 Descripción de la tabla Campo. Fuente: Elaboración Propia.	46
Tabla 6 Descripción de la tabla Campo_valor. Fuente: Elaboración Propia.....	47
Tabla 7 Descripción de la tabla Usuario_campo. Fuente: Elaboración Propia.	48
Tabla 8 Descripción de la tabla Archivo_multimedia. Fuente: Elaboración Propia.....	49
Tabla 9 Descripción de la tabla Tipologias_am. Fuente: Elaboración Propia.	49
Tabla 10 Descripción de la tabla Historial. Fuente: Elaboración Propia.....	50
Tabla 11 Muestra utilizada para el análisis de los datos de la tabla Usuario_campo. Fuente: Plataforma VideoWeb.....	51
Tabla 12 Muestra utilizada para el análisis de los datos de la tabla Campo_valor. Fuente: Plataforma VideoWeb.....	52



Índice de Figuras

Figura 1 Arquitectura 3 Capas. (Sourceforge, 2005).....	29
Figura 2 Figura 2 Arquitectura Modelo Cliente/Servidor. (Periodismo y Web 2.0, 2006)	30
Figura 3 Arquitectura Modelo Vista Controlador. (Sunsite, 2004)	31
Figura 4 Diagrama de Casos de Usos del Sistema. Fuente: Elaboración propia.	39
Figura 5 Tablas de la base de datos de la plataforma VideoWeb utilizadas en el desarrollo del algoritmo. Fuente: Plataforma VideoWeb.	46
Figura 6 Funcionalidad getContenido.Fuente: Elaboración propia.	53
Figura 7 Funcionalidad getContenidoAux.Fuente: Elaboración propia.	56
Figura 8 Funcionalidad getContenidoPersistente.Fuente: Elaboración propia.	57
Figura 9 Funcionalidad inHistorial. Fuente: Elaboración propia.	58
Figura 10 Funcionalidad getContenidoAleatorio. Fuente: Elaboración propia.	58
Figura 11 Muestra sobre los resultados luego de la personalización.Fuente: Elaboración propia.....	59
Figura 12 Muestra sobre la selección de la cantidad de recomendaciones que desea el usuario. Fuente: Elaboración propia.	60
Figura 13 Agregar nueva tipología. Fuente: Elaboración propia.	60
Figura 14 Agregar nueva multimedia. Fuente: Elaboración propia.	61
Figura 15 Agregar nuevos campos de la tipología. Fuente: Elaboración propia.	61



Introducción

En los últimos años el auge y crecimiento de Internet, tanto en cantidad de información como de usuarios, ha convertido la red en una herramienta tecnológica muy poderosa, influyendo en el conocimiento, el trabajo y el entretenimiento de millones de personas a nivel mundial. Con el esparcimiento de Internet, a los usuarios se les hace difícil encontrar los contenidos que se ajustan a sus gustos y preferencias, por esto proporcionarles información de acuerdo a sus necesidades es un reto que se vuelve cada día más complejo debido al progreso de la información disponible en la web.

Como parte de los esfuerzos realizados para brindar soluciones más efectivas a los usuarios, en la última década se han emprendido investigaciones para aprender sobre su comportamiento, con el propósito de brindarles información pertinente a sus intereses. Muchas compañías con aplicaciones web en la red de redes, invierten grandes sumas de dinero en publicidad orientadas a captar la atención del cliente, para ello hacen uso de la personalización, la cual posibilita reconocer, comprender y satisfacer a un cliente realizando productos adaptados a su contexto.

La personalización se ha convertido en un campo ampliamente investigado, en el cual se han implementado una gran variedad de herramientas para proporcionar información de forma personalizada en base a los patrones de comportamiento y particularidades del usuario con el fin de aumentar su constancia y conseguir que el mismo se sienta identificado con el sitio. (W, Kim, 2002)

Cuba también ha evolucionado en estos aspectos, por lo cual no se ve ajena a los cambios que sufre Internet en el proceso de personalización, en la Universidad de las Ciencias Informáticas (UCI) y muy en particular en el proyecto VideoWeb, perteneciente al centro GeySed de la facultad 6.

La plataforma VideoWeb es un sistema informático que permite la transmisión de video y audio digital utilizando tecnología streaming, la cual permite la visualización de estos contenidos por la red en tiempo real, brinda servicios de video y audio e incorpora gestores de usuarios y de contenido audiovisual, tiene como objetivo general entregar a cualquier institución que lo necesite una solución tecnológica integral que permita la transmisión, administración y publicación de contenido multimedia en la web, utilizando tecnologías libres.



En un inicio se comprobó que la interfaz de la plataforma no resulta intuitiva y que tampoco se muestran personalizadas las distintas acciones realizadas en la misma por la mayoría de los usuarios.

Atendiendo a los aspectos valorados el presente trabajo está enfocado a resolver el siguiente **problema científico**: La interfaz de la Plataforma VideoWeb no se muestra a los usuarios acorde a sus gustos y preferencias, lo que trae consigo un incremento de la navegación de los usuarios para encontrar sus contenidos preferidos.

Con vista a la solución del problema el **objeto de estudio** propuesto es: las técnicas y modelos de personalización para la extracción y procesamiento de información a partir de la base de conocimiento del proyecto VideoWeb.

Para solucionar este problema se tiene como **objetivo general** desarrollar un algoritmo que permita la personalización automática de la interfaz de la Plataforma VideoWeb a partir de las preferencias del usuario.

Se plantea entonces como **idea a defender**: La implementación del algoritmo propuesto permitirá la personalización automática de la interfaz de la Plataforma VideoWeb a partir de las preferencias del usuario.

Como **campo de acción** se tienen las técnicas de extracción y procesamiento a emplear en la personalización de la interfaz de la plataforma de VideoWeb.

Para darle cumplimiento al objetivo general se trazaron las siguientes **tareas de investigación**:

- Analizar y caracterizar las técnicas para la extracción y procesamiento de información a partir de bases de conocimiento.
- Definir la base de conocimientos a partir de la cual extraer la información de las preferencias del usuario.
- Seleccionar las técnicas a emplear en la personalización de la interfaz de la Plataforma VideoWeb.
- Identificar las funcionalidades que debe brindar el algoritmo.
- Desarrollar el algoritmo.
- Desarrollar un prototipo funcional para probar el algoritmo desarrollado.



- Realizar pruebas al algoritmo a través del prototipo funcional.

Con el correcto cumplimiento de las tareas se espera obtener los siguientes **resultados**:

- Documentación que recoja el resultado de la investigación realizada.
- Algoritmo para la personalización automática de la interfaz de la Plataforma VideoWeb.
- Prototipo funcional que demuestre el funcionamiento del algoritmo desarrollado.

Para adquirir los conocimientos necesarios que permitan el cumplimiento del objetivo trazado y desempeñar las tareas planeadas, se lleva a cabo una investigación en la que se utilizan los siguientes métodos científicos:

- **Analítico-Sintético:** se utiliza en el análisis de los elementos bibliográficos, definiciones y apuntes para la recopilación de información sobre las técnicas de extracción y procesamiento de información a partir de bases de conocimiento.
- **Histórico-Lógico:** ayuda a caracterizar las soluciones existentes estudiando su evolución y facilitando aprovechar puntos en común y conceptos teóricos que sean de relevancia para la realización del algoritmo.
- **Modelación:** brinda la posibilidad de crear abstracciones para explicar la realidad existente haciéndose visible en el trabajo al crear el modelo de implementación del algoritmo y un prototipo funcional que demuestre el funcionamiento del mismo.



CAPÍTULO 1. “Fundamentación Teórica”

Introducción

En el presente capítulo se registran los principales conceptos asociados al proceso de personalización, se describen las técnicas de extracción y procesamiento de contenidos que se estudiarán para obtener la información precisa para la representación de las preferencias del usuario que están reflejadas en la base de datos de la plataforma.

1. Conceptos asociados al dominio del problema.

El proceso de extraer y procesar la información almacenada a partir de las visitas a un sitio web consta de un conjunto de procesos a los cuales están asociados varios conceptos que son necesarios explicar, para lograr un mayor alcance del problema.

Información: es un conjunto organizado de datos, que constituye un mensaje sobre cierto fenómeno que permite resolver problemas y tomar decisiones, ya que su uso racional es la base del conocimiento. (Definición, 2008)

Usuario/cliente: es la persona que utiliza o trabaja con algún objeto o que es destinataria de algún servicio público, privado, empresarial o profesional. (Jimmy Wales, 2001)

La personalización de contenidos: es una técnica que trata de eliminar la sobrecarga de información mediante la adaptación de los contenidos para cada tipo de usuario. (Jimmy Wales, 2001)

Extracción de la información: es un tipo de información, cuyo objetivo es extraer automáticamente información estructurada o semi-estructurada desde modelos legibles por una computadora. (Jimmy Wales, 2001)

Minería de Datos: consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos. (Jimmy Wales, 2001)



Actualmente la personalización es una práctica muy presente y cada vez más usada, pues se utiliza en casi todos los sistemas donde se manipulen contenidos de cualquier tipo ya sean noticias, videos o imágenes, con el objetivo de satisfacer las necesidades de los clientes. La mayoría de las aplicaciones webs existentes en la actualidad están desarrolladas bajo la definición de este proceso que es la clave del éxito para cualquier sitio web en el mercado electrónico.

1.1 Principios de la personalización.

La personalización mediante reglas de negocio y los sofisticados algoritmos utilizados en la recomendación de productos mediante filtrado colaborativo de información serán los protagonistas junto a la enumeración de los principales fabricantes de software y los aspectos clave a considerar a la hora de diseñar una estrategia de personalización para un sitio web. (Fcom, 2008)

- Identificar claramente los objetivos de la personalización.
- Aprender de cada movimiento del usuario.
- Contemplar todas las posibles fuentes de información sobre el usuario.
- Obtener el máximo de información y de la forma menos intrusiva posible.
- Permitir a los usuarios comunicar que la personalización ha fallado.
- Hacer ver al usuario que se está personalizando y que el servicio le favorece a él mismo.
- Conocer lo que no le gusta al usuario es tan importante como conocer lo que le gusta.
- Nunca dejar a un usuario esperando a una respuesta personalizada, para eso es mejor no personalizar.
- Medir el éxito de la personalización.

1.1.1 Variables para el proceso de personalización.

Para poder personalizar de la forma más versátil posible, el modelo de datos del proceso de personalización se construye sobre tres tipos de variables: variables demográficas, de sesión y de operaciones para abarcar toda la información necesaria sobre el usuario. (World.com, 2004)

Las variables demográficas: representan información relacionada con características propias del cliente como persona física, tales como su nombre, apellidos, dirección, teléfono, fecha de nacimiento, sexo, etc. (World.com, 2004)



Las variables de sesión: representan información sobre como interactúa el usuario con el sitio web. Para la definición de este tipo de variables resulta imprescindible conocer con detalle todos los elementos posibles de interacción que ofrezca la interfaz de usuario, ya que es una potencial fuente de información. Con ella será posible conocer hábitos y preferencias de los clientes en base a la utilización que hace de los contenidos, los servicios y la interfaz de usuario. (World.com, 2004)

Las variables de operaciones: representan información sobre el conjunto de operaciones a nivel de transacción que el cliente ha efectuado durante su historial en el sitio web. La información suministrada por este conjunto de variables es fundamental a la hora de diseñar esquemas de personalización basados en las navegaciones anteriores, hábitos o preferencias. (World.com, 2004)

1.2 Descripción de la situación problemática.

La Plataforma VideoWeb es una solución web destinada a la gestión y transmisión de contenido audiovisual a través de la red de datos mediante el uso de tecnología Streaming (permite la transmisión de contenido multimedia a través de Internet asegurando una velocidad en la transmisión que posibilita la visualización del archivo seleccionado al mismo tiempo que se realiza la descarga). Esta es desarrollada por estudiantes y profesores del Departamento de Señales Digitales, perteneciente al centro GeySed de la facultad 6 de la Universidad de las Ciencias Informáticas empleando para su construcción herramientas y tecnologías libres. Maneja información asociada a los contenidos audiovisuales, que puede ser utilizada para enriquecer las publicaciones y ser útil a los usuarios finales.

La plataforma puede ser utilizada con fines docentes, informativos o de entretenimiento, en base a ello el producto está dirigido a entidades como universidades, ministerios, hoteles, de manera general todo el que tenga implementado una red de datos.

En la actualidad la interfaz con que cuenta la plataforma se muestra de igual manera para todos los usuarios, sin tener en cuenta sus gustos, preferencias y necesidades. Las publicaciones y demás contenidos que se muestran en portada son los mismos para todos. Este espacio tan importante podría ser utilizado para hacer énfasis en los contenidos que realmente le interesan al usuario, reduciendo así la navegación requerida para encontrar la información de su interés.



1.3 Objeto de estudio.

1.3.1 Técnicas para solicitar la información al usuario.

Los sistemas de recomendación forman parte de un tipo específico de técnica de filtro de información, los cuales presentan distintos temas de información (películas, música, libros, noticias, imágenes, etc.) que son del interés de un usuario en particular. Generalmente, un sistema de recomendación compara el perfil del usuario con algunas características de referencia de los temas, y busca predecir la ponderación que el usuario le daría a un ítem que aún el sistema no ha considerado. Estas características pueden basarse en la relación o acercamiento del usuario con el tema o en el ambiente social del mismo usuario.

Los sistemas de recomendación requieren de una interfaz de usuario que pueda determinar inteligentemente el interés de un usuario y utilizar esta información para hacer sugerencias, para esto existen dos técnicas para solicitar la información al usuario.

1.3.1.1 Calificaciones explícitas (*Explicit ratings*).

Usar calificaciones explícitas es la solución más común, donde los usuarios dicen al sistema que opinan sobre algún objeto (ej. un CD de música) o de una parte de la información que ven (ej. un artículo en un diario electrónico). Las calificaciones explícitas son bien entendidas y bastante precisas. Sin embargo, hacer una pausa para calificar explícitamente la información recibida puede alterar los patrones normales de navegación y lectura de los usuarios. (Mario Claypool, 2001)

1.3.1.2 Calificaciones implícitas (*Implicit ratings*).

Este método consiste en adquirir las calificaciones por un método distinto al de obtenerlas directamente del usuario, al igual que registrar las acciones del usuario en el navegador Web, acciones como: el tiempo de visita en una página, abrir un enlace o borrar una recomendación.

Estos indicadores de interés implícitos evitan el trabajo al usuario de realizar las calificaciones explícitas y cada acción del usuario en el sistema puede contribuir a una calificación implícita. (Morfeo Project, 2009)

Para el análisis de la base de datos de la plataforma VideoWeb se utilizó la técnica de calificaciones implícitas debido que la información registrada en la base de datos se



obtiene de la interacción de los usuarios con el sitio y esta será la información utilizada para la realización del algoritmo una vez generadas las preferencias de los usuarios.

1.3.2 Técnicas de extracción de información.

El objetivo fundamental de una aplicación que pretenda ser sensible al contexto de cada cliente es adquirir toda la información (o la mayor posible) para poder ofrecer una interfaz e información personalizada, ya que el objetivo final es brindarle al usuario los recursos más útiles y más adaptados a su contexto. Para ello será necesario estudiar las técnicas de extracción de información que permitan obtener el conocimiento preciso sobre las necesidades y preferencias de los usuarios.

1.3.2.1 Reconocimiento de nombres de entidades (*Named entity recognition*).

El Reconocimiento de Nombres de Entidades (RNE) constituye la tarea base de todo Sistema de Extracción de Información, a la vez que desempeña un papel muy importante en la búsqueda automática de respuestas, la categorización de textos, entre otras. El RNE puede definirse como la tarea de identificar y clasificar en un documento textual expresiones que identifican instancias de conceptos relevantes para algún dominio de aplicación.

El RNE no es una tarea trivial, pues resulta intratable verificar la aparición de los elementos de una lista de nombres en cada documento que se procesa. También es necesario tener en cuenta la fuerte ambigüedad que existe entre diferentes tipos de nombres de entidades, así como la influencia de fenómenos lingüísticos tales como la metonimia¹, la polisemia², las elipsis³, etc. (Yunior Ramírez Cruz , 2008)

¹ **Metonimia:** la metonimia no designa el objeto referido, sino el referente, el otro que guarda relación con él; dicha relación entre los dos términos, presente y ausente, no es de semejanza, como en la metáfora, sino de contigüidad, y así se usa la parte por el todo, el continente por el contenido, el lugar de origen por el objeto producido, etc.

² **Polisemia:** una palabra es polisémica cuando podemos expresar con ella varios significados. La polisemia se distingue de la homonimia en que se trata de una relación entre los dos planos del signo lingüístico: los diferentes significados de una palabra tienen, o han tenido, un origen común.

³ **Elipsis** es una figura retórica que omite alguno de los elementos de una frase con el objetivo de conseguir un mayor énfasis.



1.3.2.2 Análisis de texto (*Text analytics*).

El análisis de texto es un proceso cuyo objetivo es extraer automáticamente la información semi-estructurada o no estructurada de los documentos de texto. El análisis de texto surgió como la disciplina que intenta explotar los contenidos en medios electrónicos para obtener nueva inteligencia sobre distintos fenómenos que tienen al texto como un componente fundamental.

Si bien su utilidad es evidente, hacer análisis sobre un texto requiere una experiencia que va más allá del conocimiento estadístico estándar e incluso más allá del conocimiento de las herramientas usuales de la minería de datos (*data mining*). Inicialmente, disciplinas como el procesamiento del lenguaje natural, la lingüística computacional⁴, las tecnologías del habla aportaron herramientas.

Las técnicas de análisis de texto, permiten extraer contenido, clasificar texto, extraer y vincular conceptos, generar variables para incorporar en modelos predictivos, entre otras aplicaciones; pero carecen de una dimensión muy importante: la de las actitudes y emociones de aquellos que los producen, un componente fundamental para comprender el estado de los clientes. (SAS, 2010)

1.3.2.3 Minería de texto (*Text mining*).

La minería de texto puede definirse en términos generales como una herramienta capaz de abarcar una amplia gama de dominios, desde aquellos de la recuperación, extracción de información, presentación, resumen de documentos y minería de datos aplicada a textos.

Se plantea que la minería de texto consiste en la búsqueda de regularidades o patrones que se encuentran en un texto, a partir de técnicas de aprendizaje automático; por tanto, se considera como una de las muchas ramas de la lingüística computacional. (MSc. Eleazar Botta-Ferret, 2007)

Principales técnicas utilizadas en la minería de textos.

⁴Lingüística Computacional (*Computational Linguistics*) puede considerarse una disciplina de la lingüística aplicada y la Inteligencia Artificial, y tiene como objetivo la realización de aplicaciones informáticas que imiten la capacidad humana de hablar y entender. A la Lingüística Computacional se le llama a veces Procesamiento del Lenguaje Natural (PLN), o *Natural Language Processing* (NLP).



Grafos conceptuales: es un grafo bipartito que tiene dos tipos de nodos, conceptos y relaciones conceptuales. Los grafos se comparan utilizando conocimiento del dominio como diccionarios de sinónimos y jerarquías de conceptos. Se realiza una operación de intersección entre dos de ellos para dar un resumen de ambos y a dicho resumen se le valora con una puntuación que indica el grado de similitud entre ambos textos. (MSc. Eleazar Botta-Ferret, 2007)

La agrupación de dos o más grafos permite descubrir la estructura oculta de la colección de textos. Para agruparlos se pueden utilizar técnicas como las estrategias colaborativas o las técnicas de grafos conceptuales que aportan mayor semántica. Todas ellas se basan en suministrar a los algoritmos, un conjunto de ejemplos a partir de los cuales se generan las agrupaciones. (MSc. Eleazar Botta-Ferret, 2007)

Programación lógica inductiva: esta técnica permite introducir conocimiento del dominio en forma de definiciones mediante predicados relacionados. Requiere no sólo de un conjunto de entrenamiento con ejemplos sino también de las relaciones descubiertas. La ventaja que posee es su capacidad de representación basada en una lógica de segundo orden, que permite generalizar y descubrir definiciones de conceptos de forma automática. Normalmente, se utiliza *Prolog*⁵ para programar herramientas. (MSc. Eleazar Botta-Ferret, 2007)

1.3.2.4 Contexto de usuario

El contexto de usuario está compuesto de dos tipos de características, factores humanos (perfil de usuario) y factores del entorno físico (entorno de ejecución). Los factores humanos pueden dividirse en categorías: información acerca del usuario (conocimiento sobre sus hábitos, estado emocional, condiciones biofisiológicas), el entorno social del usuario (co-localización sobre los demás, interacciones sociales, dinámicas de grupo) y por último las tareas del usuario (actividades espontáneas, objetivos generales, etc.).

Creación del contexto de usuario.

El contexto de usuario es un conjunto de datos heterogéneo, que han de obtenerse de fuentes diversas, por ello, no es posible abordar la adquisición de datos para construir el

⁵ **Prolog** es un lenguaje de programación creado para representar y utilizar el conocimiento que se tiene sobre un determinado dominio. Más exactamente, el dominio es un conjunto de objetos y el conocimiento se representa por un conjunto de relaciones que describen las propiedades de los objetos y sus interrelaciones.



contexto de una manera uniforme.

Gran parte del contexto de ejecución (entorno físico) se obtendrá automáticamente de la solicitud del navegador, mediante las cabeceras HTTP (dirección IP, sistema operativo del cliente, navegador, etc.). Esta información se analizará directamente extrayendo los campos de la cabecera HTTP cuando se reciba la petición en el servidor. (Morfeo Project, 2009)

Existen algunas características del contexto de ejecución como: el conjunto de aplicaciones instaladas en el cliente, que deberán enviarse junto con cada petición, pero que se pueden extraer del cliente y ser enviadas al servidor de manera automática junto con las peticiones que se realicen (es decir, son datos objetivos y fácilmente medibles, por lo que no se requiere ningún proceso de inferencia para obtenerlos). (Morfeo Project, 2009)

Perfil de usuario:

En el caso del perfil de usuario las características principales a considerar, son aquellas relativas a los intereses del usuario: deben devolverse recursos compatibles (o que pueden interactuar) con otros que ya esté usando el usuario y deben recomendársele aquellos que sean similares (o mejores) a otros que haya usado. Además, cobra muchísima importancia la comunidad de usuarios: deberán recomendarse o descubrirse automáticamente los recursos usados por aquellos con intereses similares.

Algunas características del perfil (nombre, edad, ocupación, etc.), han de indicarse en la plataforma al propio usuario de manera explícita (mediante alguna herramienta proporcionada por la plataforma). Estas características también son datos objetivos que no necesitan un proceso de inferencia posterior.

Sin embargo, las características más útiles del perfil de usuario (intereses, aficiones, hábitos, estado emocional, entorno social, etc.) son las más complejas de obtener y no son valores objetivos, por lo que es necesario determinar de qué valores objetivos se pueden inferir todas estas características, así como qué proceso ha de seguirse para llegar a un conocimiento útil de los aspectos subjetivos de un usuario. Para realizar esta tarea, habría de analizarse primero qué técnicas de extracción de conocimiento se han aplicado tradicionalmente en la Ingeniería del Conocimiento y la Minería de Datos, estudiando además la aproximación social a la extracción de conocimiento. (Morfeo Project, 2009)



Conclusiones Parciales del Capítulo 1

En este capítulo se describen todos los conceptos asociados al dominio del problema y las técnicas de extracción de información para poder personalizar la interfaz de la plataforma VideoWeb. Dentro de las técnicas estudiadas las más factibles para trabajar con la base de conocimiento del proyecto serían el Contexto de Usuario y el Análisis de Texto. Estas técnicas permiten lograr una mejor comprensión sobre las preferencias y hábitos de los usuarios para luego con la información necesaria sobre los mismos analizar los resultados alcanzados y extraer las variables fundamentales para la realización del algoritmo.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

Introducción

En el presente capítulo se describen los conceptos asociados al funcionamiento, las acciones que debe realizar el algoritmo y el prototipo funcional, también se estudiarán las herramientas necesarias para la implementación de ambos, así como las variables que fueron extraídas de la base de datos para lograr el correcto desarrollo del algoritmo a partir de la información suministrada por los usuarios.

2.1 Arquitectura.

Con el objetivo de comprender el desarrollo de cualquier software, se hace necesario describir la arquitectura definida. La arquitectura de software a grandes rasgos es una vista del sistema que incluye los componentes principales del mismo, la vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

2.1.1 Arquitectura en 3 Capas.

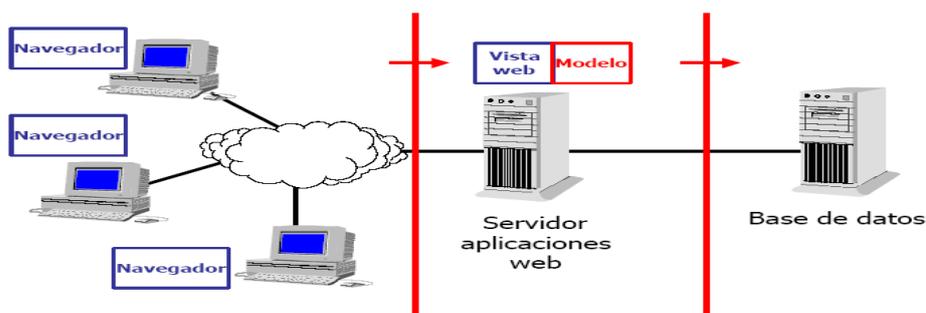


Figura 1 Arquitectura 3 Capas. (Sourceforge, 2005)

La arquitectura 3 capas o programación 3 capas consiste literalmente en separar un proyecto en Capa de Presentación, Capa de Negocio y Capa de Datos. Esto permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

Ventajas de esta Arquitectura:



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

- ✓ El desarrollo se puede llevar a cabo en varios niveles.
- ✓ Desarrollos paralelos (en cada capa).
- ✓ Aplicaciones más robustas debido al encapsulamiento.
- ✓ En caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.
- ✓ Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica).
- ✓ Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).
- ✓ Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buena escalabilidad, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad. (Kernelerror, 2009).

2.1.2 Modelo Cliente/Servidor.

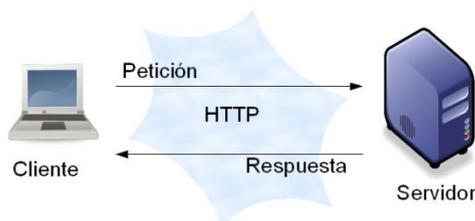


Figura 2 Arquitectura Modelo Cliente/Servidor. (Periodismo y Web 2.0, 2006)

En el modelo Cliente/ Servidor presenta las siguientes características:

- ✓ El Cliente y el Servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades o tareas independientes. Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma. Un servidor da servicio a múltiples clientes en forma concurrente. Cada plataforma puede ser escalable independientemente. Los cambios realizados en las plataformas de los Clientes o de los Servidores, ya sean por actualización o por reemplazo tecnológico, se realizan de una manera transparente para el usuario final. La interrelación entre el hardware y el software están basados en una infraestructura poderosa, de tal forma que el acceso a los recursos de la red no muestra la complejidad de los diferentes tipos de formatos de datos y de los protocolos.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

- ✓ Un sistema de servidores realiza múltiples funciones al mismo tiempo que presenta una imagen de un sólo sistema a las estaciones clientes. Esto se logra combinando los recursos de cómputo que se encuentran físicamente separados en un solo sistema lógico, proporcionando de esta manera el servicio más efectivo para el usuario final. Además, se constituye como el nexo de unión más adecuado para reconciliar los sistemas de información basados en mainframes o minicomputadores, con aquellos otros sustentados en entornos informáticos pequeños y estaciones de trabajo. Designa un modelo de construcción de sistemas informáticos de carácter distribuido.

En conclusión, Cliente/Servidor puede incluir múltiples plataformas, bases de datos, redes y sistemas operativos. Estos pueden ser de distintos proveedores, en arquitecturas propietarias y no propietarias y funcionando todos al mismo tiempo. (Kernelerror, 2009)

2.1.3 Arquitectura Modelo Vista Controlador (MVC).

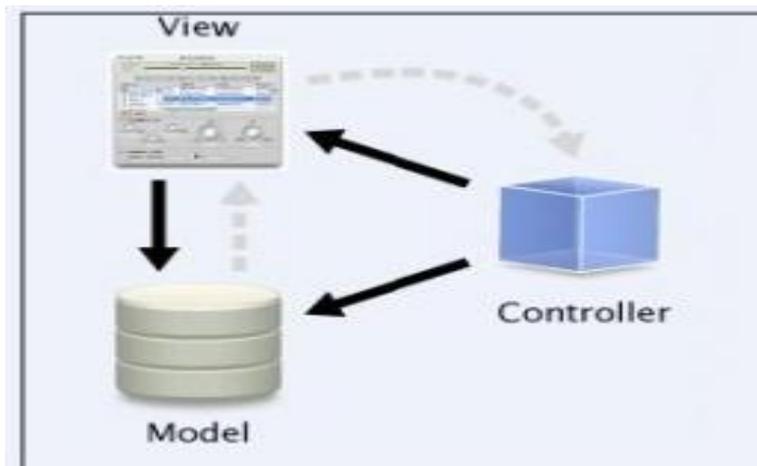


Figura 3 Arquitectura Modelo Vista Controlador. (Sunsite, 2004)

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El estilo de llamada y retorno MVC, se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista. Sus características principales son que el Modelo, las Vistas y los Controladores se tratan como entidades separadas; esto hace que cualquier cambio producido en el Modelo se refleje automáticamente en cada una de las Vistas.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

El Modelo es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo, sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo, a través, de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo, a través, de una referencia al mismo.

Este modelo de arquitectura presenta varias ventajas:

- Existe una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.
- Consta de un API ⁶muy bien definido; aquel individuo haga uso del API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.

La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación. (Sunsite, 2004)

¿Por qué arquitectura Modelo Vista Controlador?

Llevando este concepto al marco de trabajo y una vez estudiado lo referente a las arquitecturas se determinó (por las características que presenta) utilizar la arquitectura MVC, ya que ésta separa la lógica de negocio (el modelo) y la presentación (la vista), utilizado la misma se consigue un mantenimiento más sencillo de la aplicación. Además se tendrá una organización de código separado por sus funciones.

⁶ Una **API** (*Aplicación Programming Interface*) Interfaz de Programación de Aplicaciones es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

2.2 Lenguaje de Programación.

Durante los últimos años se ha acumulado experiencia en el diseño y utilización de los lenguajes de programación. En el año 2004 Lauden plantea que “Un lenguaje de programación es un sistema notacional para describir computaciones en una forma legible tanto para la máquina como para el ser humano”. Después de analizar el siguiente concepto se puede concluir que, un lenguaje de programación facilita la comunicación entre el programador y la computadora.

2.2.1 PHP

Lenguaje de programación denominado preprocesador de hipertexto, (del inglés “HyperText Pre-processor”), es libre y multiplataforma. Posee una amplia documentación en su página oficial posibilitando gran comprensión del mismo, se sustenta en la actualidad bajo el paradigma más difundido actualmente en el mundo que es programación orientado a objeto, incluye también la programación estructurada y servicios web. Presenta excelente integración con todos los motores de base de datos. Cuenta con una biblioteca que trae un conjunto de funciones para realizar cualquier operación (acceso a base de datos, encriptación, envío de correo, XML, creación de PDF, entre otros). PHP, es un lenguaje de programación usado normalmente para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web, es la versión libre del sistema equivalente de Microsoft ASP.

Principales ventajas de PHP.

- Sintaxis cómoda: PHP cuenta con una sintaxis similar a la de C, C++ o Perl. Lo más destacado ocurre a nivel semántico: el chequeo de tipos es poco estricto, cuando se crea una variable no se tiene que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo. Esto es muy flexible y cómodo para el desarrollador, aunque los errores que se cometen pueden ser muchos más graves y difíciles de corregir al reducirse las posibilidades del intérprete para detectar incompatibilidades entre variables.
- Soporta objetos y herencia: tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores, etc. Además soporta herencia, aunque no múltiple.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

- Ejecución en Servidor: un lenguaje del lado del servidor es aquel que se ejecuta justo antes de que se envíe la página, a través, de internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página web con el código HTML resultante de la ejecución del código PHP compatible con todos los navegadores.
- Se puede incrustar código PHP con etiquetas HTML.
- Compatibilidad con bases de datos: amplio soporte para una gran cantidad de bases de datos. Tiene acceso un gran número de gestores de bases de datos: Adabas D, dBase, Empress, Ingress, InterBase, FrontBase, DB2, Informix, MySQL, ODBC, Oracle, PostgreSQL, Sybase, etc.
- Multiplataforma: funciona tanto en sistemas Unix o Linux con servidor web Apache como en sistemas Windows con *Microsoft Internet Information Server*, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra.
- Licencia de software libre: es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, ni existen límites en su distribución, es posible ampliarlo con nuevas funcionalidades si así lo desea.
- Extensa librería de funciones: cuenta con una extensa librería de funciones que facilitan enormemente el trabajo de los desarrolladores.

2.2.2 C++

Lenguaje de programación de alto nivel orientado a objetos, que se puede utilizar para escribir software de bajo nivel, componentes de sistemas operativos como para el desarrollo rápido de aplicaciones. Los compiladores de C++ generan código con un alto grado de optimización en memoria y velocidad, lo que lo convierte en uno de los lenguajes más eficientes. Es un lenguaje que está estandarizado y un mismo código fuente se puede compilar en diversas plataformas. (YLL, Shelly, 2003) Entre sus principales características se encuentran:

- El encapsulamiento: mecanismo que agrupa el código, los datos que maneja y los mantiene protegidos frente a cualquier interrupción.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

- El polimorfismo: posibilita que se utilice un mismo nombre para dos o más propósitos relacionados pero técnicamente diferentes.
- La herencia: mediante la herencia un objeto puede obtener un conjunto de propiedades de otro objeto a las que se puede agregar características específicas de él.

¿Por qué PHP?

La aparición de soluciones más adecuadas y sencillas hace que PHP se convierta en la mejor opción para la implementación del algoritmo, primeramente porque presenta diferentes librerías que permitirán realizar el mismo de manera más fácil y rápida. Su uso permitirá que el algoritmo pueda ser utilizado por cualquier entidad que lo necesite sin importar el sistema operativo con el que trabaje.

2.3 Herramientas de Desarrollo.

Las herramientas CASE (*Computer Aided Software Engineering*), Ingeniería de Software Asistida por Computadora se han venido ampliando y desarrollando progresivamente. Estas herramientas son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de las mismas en términos de tiempo y de dinero. Pueden ayudar en todos los aspectos del ciclo de vida del software en tareas como: realizar un diseño del proyecto, cálculo de costes, implementación del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

2.3.1 Visual Paradigm.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite modelar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación.

2.3.2 Rational Rose.

Es una herramienta software para el modelado visual mediante UML de sistemas software. Permite especificar, analizar y diseñar el sistema antes de codificarlo.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

Algunas de las características que presenta Rational Rose son:

- ✓ Mantiene la consistencia de los modelos del sistema software.
- ✓ Chequeo de la sintaxis UML.
- ✓ Generación de documentación automáticamente.
- ✓ Generación de código a partir de los modelos.
- ✓ Ingeniería inversa (crear modelo a partir código).

¿Por qué Visual Paradigm?

Se decidió el uso de Visual Paradigm, pues su uso está muy estandarizado a nivel mundial, además constituye una herramienta multiplataforma madura y acabada, también permite realizar los diagramas necesarios para el desarrollo del algoritmo con la calidad requerida.

2.4 Variables obtenidas del trabajo con la base de datos de la plataforma.

El diseño de la interfaz centrada en el usuario se caracteriza por asumir que todo el proceso de diseño del sitio debe estar conducido por el usuario, sus necesidades, características y objetivos. Centrar el diseño en los clientes implica involucrarlos en el proceso de desarrollo del sitio; conocer cómo son, qué necesitan y para qué usan el sitio.

Toda la información obtenida de los estudios realizados a la base de datos sobre el comportamiento de los usuarios en la plataforma modeló un conjunto de variables comunes entre ellos, tales como necesidades de información y conocimientos. Estas variables son generadas a partir de las tipologías brindadas por la plataforma enumeradas en la siguiente tabla.

Tabla 1 Variables obtenidas de la base de datos. Fuente: Elaboración propia.

Tipologías				
Películas	Docentes	Videos	Documentales	Series
Género	Exponente	Cantante	Género	Reparto



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

Director	País	Álbum	País	Género
Reparto	Idioma	Editor		País
Escritor		Productor		Idioma
País				

La base de datos de la plataforma VideoWeb cuenta con una serie de tablas que contienen datos asociados a los diferentes tipos de contenidos audiovisuales, en el caso de las películas se registran campos como el título de la misma, el año, el director, el género al que pertenece entre otros campos, de ellos se pudo constatar que algunos mantenían el mismo valor para los diferentes contenidos audiovisuales los cuales al repetirse pueden generar preferencias de usuario.

Estas variables obtenidas durante el análisis realizado a la base de datos de la Plataforma VideoWeb fueron extraídas de los campos asociados a los materiales audiovisuales que podían generar preferencias de usuario al ser comunes para varios materiales, las mismas constituyen la base de conocimientos a utilizar para el desarrollo del algoritmo. Además, se puede mencionar, que en la personalización también pueden influir las secciones y categorías en las cuales están ubicados los contenidos una vez publicados en la Plataforma, por ejemplo: todas las películas publicadas en la categoría “Cubanas” dentro de la sección “Películas”.

2.5 Requisitos Funcionales

Los requisitos funcionales definen las acciones que debe realizar el sistema. Son capacidades o condiciones que el sistema debe cumplir. (Falgueras, 2003). A partir de un estudio realizado se definieron los requisitos funcionales que debe cumplir el algoritmo para personalizar correctamente la interfaz de la plataforma los cuales son enumerados a continuación:

RF1 Personalizar interfaz de usuario.

El sistema es el encargado de personalizar la interfaz de un usuario partiendo del comportamiento manifestado durante las sesiones de navegación más visitadas por el mismo, ejemplo: los videos que más le gustan, la serie de su preferencia.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

RF2 Notificar nuevo contenido.

El revisor debe hacerle llegar al usuario notificaciones (por correo o, a través, de su perfil) del nuevo contenido multimedia publicado en el entorno y el usuario lo añade o no dentro de sus gustos y preferencias.

2.6 Requisitos no Funcionales

Los requerimientos no funcionales forman una parte significativa de la especificación. Son propiedades o cualidades que el producto debe tener. (Falgueras, 2003). Estos requerimientos son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, puede marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. Los requisitos no funcionales que debe cumplir el algoritmo son:

2.6.1 Restricciones en el diseño y la implementación.

- El patrón arquitectónico que se debe emplear en el desarrollo del prototipo funcional es el Modelo-Vista-Controlador.
- El lenguaje de programación que se debe utilizar es PHP para la implementación del algoritmo.
- Para la modelación de la arquitectura se utilizará el Visual Paradigm.

2.7 Definición de los Actores del Sistema.

Un actor no es parte del sistema, es un rol de un usuario que puede intercambiar información y puede representar un ser humano, a un software o a una máquina que interactúa con el sistema. (Pressman, 1988). En el desarrollo del algoritmo se definieron dos actores: el usuario registrado y el revisor que en un momento determinado efectuarán un grupo de acciones en el sistema, los mismos se describen a continuación:



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

Tabla 2 Descripción de los actores del sistema. Fuente: Elaboración propia.

Actor	Descripción
Usuario registrado	Tiene privilegios para autenticarse en el sistema y realizar búsquedas de contenidos multimedia dentro de la plataforma.
Revisor	Tiene privilegios para autenticarse en el sistema y publicar nuevos contenidos multimedia dentro de la plataforma.

2.8 Diagrama de Caso de Uso del Sistema (DCUS).

El Diagrama de casos de uso del sistema, define las relaciones entre los actores y los casos de uso. Para la realización del algoritmo se definieron un conjunto de acciones que deben ser ejecutadas por distintos actores y que desencadenan un conjunto de operaciones. Las relaciones entre las acciones y los actores son agrupadas en el diagrama de casos de uso del sistema que se muestra a continuación.

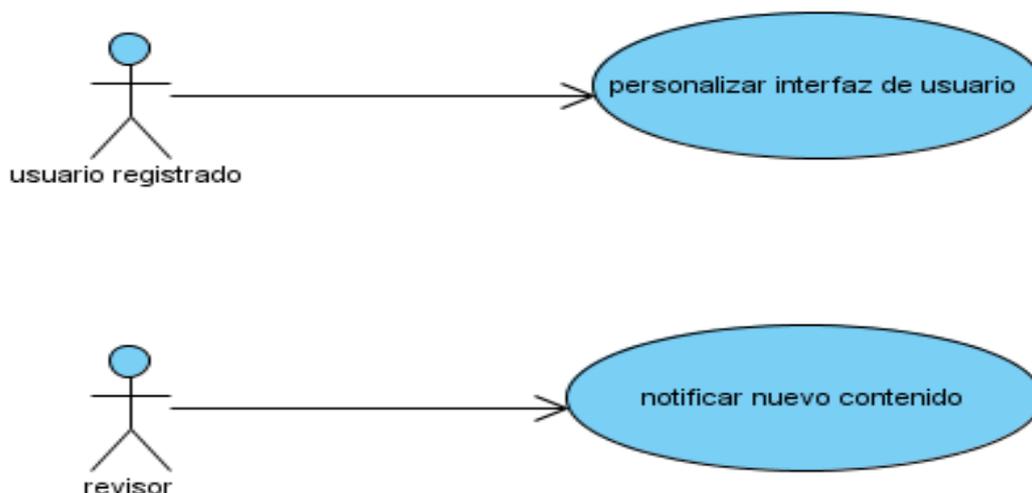


Figura 4 Diagrama de Casos de Usos del Sistema. Fuente: Elaboración propia.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

2.9 Descripción de los Casos de Usos.

Tabla 3 Personalizar interfaz de usuario. Fuente: Elaboración propia.

Caso de Uso:	Personalizar interfaz de usuario
Actores:	Usuario registrado
Propósito	Este caso de uso se lleva a cabo con el objetivo de personalizar la interfaz de la plataforma VideoWeb para el usuario que inicia sesión a partir de sus preferencias.
Resumen:	El caso de uso comienza cuando se autentica un usuario en la plataforma VideoWeb, la que es personalizada ubicando en el tope de publicaciones las que pueden ser de preferencia para el usuario que se autentica.
Precondiciones:	Debe existir al menos un usuario registrado en la plataforma. Debe estar publicado al menos un contenido multimedia.
Referencias	RF1
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El caso de uso se inicia cuando el usuario introduce sus datos de acceso a la plataforma VideoWeb.	2. El sistema realiza el análisis del historial de navegación del usuario que se autentica con el objetivo de obtener, en base a los accesos anteriores, las listas de contenidos audiovisuales que más coinciden con su preferencia. 3. El sistema ordena la lista de los contenidos publicados y que son de posible preferencia por el usuario, colocando en las primeras posiciones los que no han sido visualizados por este y de más reciente publicación. 4. Se muestran, en la región de últimas publicaciones de la portada, los contenidos obtenidos en la lista.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

	Culmina así el caso de uso.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
	2. En caso de que el usuario que se autentica no posea historial previo de navegación, se muestran en la región de últimas publicaciones de la portada, los contenidos que por defecto define la plataforma. Culmina así el caso de uso.
	3. En caso de que no exista contenido multimedia publicado de la posible preferencia del usuario, se muestran en la región de últimas publicaciones de la portada, los contenidos que por defecto define la plataforma. Culmina así el caso de uso.
Poscondiciones	Se personaliza la interfaz de usuario de la plataforma VideoWeb de acuerdo a las preferencias del usuario que accede a ella.

Tabla 4 Notificar nuevo contenido. Fuente: Elaboración propia.

Caso de Uso:	Notificar nuevo contenido
Actores:	Revisor
Propósito	Informar a los usuarios de la plataforma VideoWeb sobre los nuevos contenidos publicados que puedan ser de su preferencia.
Resumen:	El caso de uso comienza con la publicación de un nuevo contenido multimedia en la Plataforma VideoWeb, esta nueva publicación es analizada para comprobar si es de la preferencia de alguno de los usuarios. El caso de uso termina con una notificación a los usuarios para los cuales la nueva publicación constituye una posible preferencia.
Precondiciones:	Debe existir al menos un usuario con historial de navegación en la aplicación. Debe estar publicado al menos un contenido multimedia.
Referencias	RF1



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<ol style="list-style-type: none"> 1. El caso de uso se inicia cuando el Revisor publica un nuevo contenido multimedia introduciendo su clasificación y los datos descriptivos asociados a este. 	<ol style="list-style-type: none"> 2. El sistema compara los datos descriptivos del nuevo contenido publicado con los almacenados en la Base de Datos, los que corresponden al historial de navegación y acceso a contenido multimedia de los usuarios en la plataforma. 3. El sistema elabora una lista de usuarios en los que existen coincidencias válidas para determinar que el nuevo contenido puede ser de su interés. 4. El sistema notifica mediante el correo o un mensaje de alerta la publicación del nuevo contenido a los usuarios de la lista obtenida. Culmina así el caso de uso.
Pos condiciones	Se les notifica a los usuarios correspondientes la publicación de nuevo contenido multimedia que puede ser de su preferencia.

2.10 Técnicas seleccionadas para el desarrollo del algoritmo.

El análisis de la información de la base de datos mediante las técnicas de extracción de conocimiento se realiza con el propósito de obtener información que permita adquirir el contenido capaz de generar preferencias de usuario. Un profundo análisis sobre las técnicas de extracción de información permitió reconocer como las más factible para lograr personalizar los servicios de la Plataforma VideoWeb, a las técnicas Contexto de usuario y el Análisis de texto, ambas técnicas en su conjunto permiten elaborar todos los datos necesarios para desarrollar correctamente el algoritmo, así como responder a los problemas de comprensión sobre las necesidades de los clientes en la web.

El Análisis de texto permite extraer y generar un conjunto de variables que son fundamentales para la implementación del algoritmo. Esta técnica permite estudiar y formalizar la estructura de



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

los textos en sus diversos niveles, es decir, los contenidos tienen una estructura compleja que se debe conocer apropiadamente para poder explotar la información valiosa contenida en ellos, implícita o explícitamente. También permite la recuperación de información, el reconocimiento de patrones dentro del texto, teniendo como objetivo principal convertir texto en datos precisos que sean capaces de generar preferencias de usuario.

Por otra parte, el Contexto de usuario es la técnica fundamental para personalizar la interfaz de usuario de la plataforma VideoWeb, es la encargada de acumular todos los factores humanos y de entorno físico que brindan los clientes que navegan en la web. La utilización de esta técnica para la implementación del algoritmo permite tener en cuenta la importancia del usuario en el sistema, analizando quién es, qué es lo que prefiere de los contenidos que brinda la plataforma, además permite obtener la información pertinente a las necesidades de los usuarios, teniendo en cuenta aspectos como: las características personales de los clientes, sus preferencias, su historial en la base de datos de la plataforma.



CAPÍTULO 2. “Características del algoritmo para la personalización de la plataforma VideoWeb”

Conclusiones Parciales del Capítulo 2.

En este capítulo se realizó una descripción sobre las funciones que debe realizar el algoritmo. La descripción de los casos de uso permitió sentar las bases para la implementación del algoritmo de personalización de la interfaz de la plataforma y del prototipo funcional que debe validar el funcionamiento de dicho algoritmo. El estudio de las técnicas de extracción de conocimiento permitió seleccionar de manera correcta las técnicas que deberán ser utilizadas durante el desarrollo del algoritmo y luego de un análisis realizado a un grupo de tecnologías y herramientas estudiadas se seleccionaron las más factibles para realizar correctamente el algoritmo y el prototipo funcional.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Introducción

El desarrollo del presente capítulo está enfocado a la implementación del algoritmo y del prototipo funcional a partir de la descripción de los casos de usos que se trataron en el capítulo anterior. Se definen las tablas de la base de datos a utilizar para la implementación del algoritmo. Además de realizar un análisis de los resultados alcanzados con el desarrollo del algoritmo, a través, de las pruebas realizadas con del prototipo funcional.

3.1 Desarrollo del algoritmo para la personalización de la interfaz de la plataforma VideoWeb.

Para el desarrollo del algoritmo se tuvo en cuenta la estructura de la base de datos, a partir de un análisis realizado se seleccionaron las tablas que son relevantes para la obtención de preferencias de usuario, las cuales se numeran y describen a continuación:

3.1.1 Listado de las tablas utilizadas para la implementación del algoritmo de personalización.

- campo.
- campo_valor.
- usuario_campo.
- archivo_multimedia.
- tipologias_am.
- historial.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

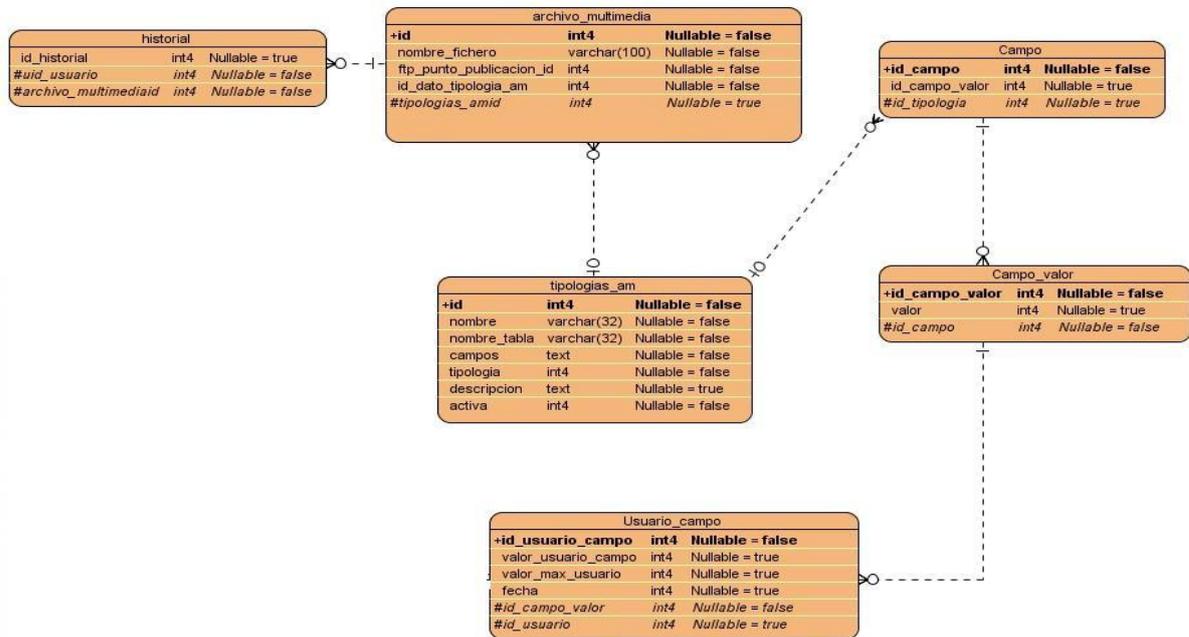


Figura 5 Tablas de la base de datos de la plataforma VideoWeb utilizadas en el desarrollo del algoritmo. Fuente: Plataforma VideoWeb.

Descripción de las tablas y atributos seleccionados para el desarrollo del algoritmo.

A continuación se describen de forma general los datos que se almacenan en las tablas seleccionadas anteriormente para la implementación del algoritmo, donde se explica brevemente la información que representan los atributos de cada una de ellas, justificando la selección de las mismas.

Tabla 5 Descripción de la tabla Campo. Fuente: Elaboración Propia.

Nombre: campo		
Descripción: En esta tabla se almacenan los campos relevantes en la obtención de preferencias de usuario.		
Atributo	Tipo	Descripción
id_campo	int4	Valor numérico único y auto incremental que identifica a cada



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

		uno de los campos de esta tabla.
id_tipologia	int4	Es la variable encargada de identificar la tipología del archivo multimedia.
id_campo_tipologia	int4	Es la variable encargada de identificar el campo perteneciente a la tipología de archivo multimedia.

Tabla 6 Descripción de la tabla Campo_valor. Fuente: Elaboración Propia.

Nombre: campo_valor		
Descripción: En esta tabla se almacenan los valores que toman los campos relevantes en la obtención de preferencias de usuario.		
Atributo	Tipo	Descripción
id_campo_valor	int4	Valor numérico único y auto incremental que identifica a cada uno de los campos de esta tabla.
id_campo	int4	Es la variable encargada de identificar los campos relevantes para la obtención de preferencias.
Valor	int4	Valor a tomar por el campo para obtener las preferencias.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Tabla 7 Descripción de la tabla Usuario_campo. Fuente: Elaboración Propia.

Nombre: usuario_campo		
Descripción: En esta tabla se almacenan los valores que toman los campos relevantes en la obtención de preferencias de usuario.		
Atributo	Tipo	Descripción
id_usuario_campo	int4	Valor numérico único y auto incremental que identifica a cada uno de los campos de esta tabla.
id_usuario	int4	Identificador de los usuarios.
id_campo_valor	int4	Identificador del valor que toman los campos relevantes para la obtención de preferencias de usuarios.
valor_usuario_campo	int4	Valor de preferencia del usuario por el campo.
valor_max_usuario	int4	Valor máximo histórico del usuario por el campo.
Fecha	int4	Fecha en que se registra.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Tabla 8 Descripción de la tabla Archivo_multimedia. Fuente: Elaboración Propia.

Nombre: archivo_multimedia		
Descripción: En esta tabla se almacenan los datos de las publicaciones de archivos multimedia que posee el sistema.		
Atributo	Tipo	Descripción
id	int4	Valor numérico único y auto incremental que identifica a cada uno de los archivos multimedia que posee el sistema.
nombre_fichero	varchar(100)	Nombre que posee la publicación para ser mostrada.
ftp_punto_publicacion_id	int4	Id del servicio de almacenamiento donde se va a guardar el archivo multimedia.
id_tipologia_am	int4	Id de la tipología a la que pertenece el archivo multimedia.
id_dato_tipologia_am	int4	Id del dato del archivo según la tipología.

Tabla 9 Descripción de la tabla Tipologias_am. Fuente: Elaboración Propia.

Nombre: tipologias_am
Descripción: En esta tabla se almacenan los datos de las tipologías de archivo multimedia que posee el sistema.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Atributo	Tipo	Descripción
id	int4	Valor numérico único y auto incremental que identifica a cada una de las tipologías de archivo multimedia que posee el sistema.
nombre	varchar(32)	Nombre de la tipología.
nombre_tabla	varchar(32)	Nombre de la tabla en la base de datos.
campos	Text	Arreglo con la estructura de cada uno de los campos que conforman la tipología.
tipologia	int4	Identifica si es una tipología que tiene asociado video o no.
descripcion	Text	Breve descripción de la tipología.
activa	int4	Identifica si fue creada o no la tipología.

Tabla 10 Descripción de la tabla Historial. Fuente: Elaboración Propia.

Nombre: historial		
Descripción: En esta tabla se almacenan los historiales de acceso a los archivos multimedia.		
Atributo	Tipo	Descripción
id_historial	int4	Valor numérico único y auto incremental que identifica a cada uno



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

		de los campos de esta tabla.
id_usuario	int4	Identificador del usuario.
archivo_multimediaid	int4	Identificador del archivo multimedia.

3.1.1 Pasos realizados en el desarrollo del algoritmo basado en los sistemas de recomendación.

1. Identificar a los usuarios de forma independiente para adaptar el contenido a sus necesidades a partir de la información registrada en la base de datos.
2. Analizar los contenidos que predominan en la base de datos sobre el usuario, para lograr un mejor alcance en la implementación del algoritmo.

Tabla 11 Muestra utilizada para el análisis de los datos de la tabla Usuario_campo. Fuente: Plataforma VideoWeb.

	id_usuario [PK] serial	valor_usua int_unsign	valor_max int_unsign	id_campo_ int_unsign	id_usuario int_unsign	fecha integer
1	1	60	100	1	1	1305888927
2	2	50	80	3	1	1305888927
3	3	30	60	4	1	1305888927
4	4	20	60	5	1	1305888927
5	5	70	70	6	1	1305888927
6	6	30	30	7	1	1305888927
7	7	7	30	8	1	1305888927
8	8	30	90	9	1	1305888927
9	9	60	60	10	1	1305888927
*						



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Tabla 12 Muestra utilizada para el análisis de los datos de la tabla Campo_valor. Fuente: Plataforma VideoWeb.

	id_campo_ [PK] serial	id_campo int_unsign	valor text
1	1	1	Cuba
2	2	1	Japón
3	3	1	EEUU
4	4	1	Argentina
5	5	2	Pepe
6	6	2	Juan
7	7	2	Lolo
8	8	3	Acción
9	9	3	Comedia
0	10	3	Drama
1	11	4	Juan Forme
2	12	4	Juanes
3	13	4	Frank Delc
4	14	5	sdasdfasd
5	15	5	ppoppop
6	16	5	jjjjjjjjj
*			

Anteriormente se expuso una muestra de los datos que engloban las tablas que fueron utilizadas para el análisis de cómo funcionaría el algoritmo a partir de la información recogida en ellas.

3. Como resultado de los pasos anteriores cada vez que el cliente interactúa con la plataforma recibe una serie de contenidos que concuerdan con sus necesidades, mediante las comparaciones que realiza el algoritmo basado en sistemas de recomendaciones (un sistema de recomendación se basa en realizar comparaciones con algunas características de referencia de los temas que el usuario tiene dentro de sus gustos y preferencias buscando predecir la ponderación que el usuario le daría a un contenido determinado) con la información obtenida sobre las preferencias de los usuarios. Cuanto más interactúe mejor serán las recomendaciones que recibe y más efectiva será la personalización.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

3.2 Implementación del algoritmo para personalizar la interfaz de la plataforma VideoWeb.

3.2.1 Funcionalidades realizadas para el desarrollo del algoritmo.

Funcionalidad `getContentido`.

Es la funcionalidad principal del algoritmo, encargada de retornar las recomendaciones con los contenidos que se encuentran dentro de las preferencias de los usuarios.

```
function getContentido($idTipologia, $cant, $campos, $idUserio) {  
    $preferencias = getContentidoAux($idTipologia, $cant, $campos, $idUserio);  
    $recomendaciones = array();  
    getContentidoPersistente($idTipologia, $preferencias, $cant, $idUserio, $recomendaciones);  
  
    if (count($recomendaciones) === 0) {  
        $recomendaciones = getContentidoAleatorio($cant);  
    }  
    return $recomendaciones;  
}
```

Figura 6 Funcionalidad `getContentido`. Fuente: Elaboración propia.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Funcionalidad `getContenidoAux`.

Es la funcionalidad encargada de retornar los campos de una tipología que puedan generar preferencias sobre el usuario y retorna una lista con todos los campos que prefiere el usuario a partir de la tipología analizada.

```
function getContenidoAux($idTipologia, $cant, $campos, $idUserio) {
    if (is_array($campos)) {
        $encontrado = array();
        foreach ($campos as $campo) {
            $result = db_query("SELECT * FROM {campo_valor} WHERE id_campo=%d", $campo->id_campo);
            if ($result) {
                $valores = array();
                $encontrado[$campo->id_campo] = array();
                while ($valor = db_fetch_object($result)) {
                    $valores[] = $valor;
                }
                $encontrado[$campo->id_campo] = $valores;
            }
        }
    }
}
```

```
$valores_preferencia = array();
foreach ($encontrado as $idCampo => $campo) {
    $valores_campos = $campo;
    $valores_preferencia[$idCampo] = array();
    foreach ($valores_campos as $valor_campo) {
        $result = db_query("SELECT * FROM {usuario_campo} uc WHERE id_campo_valor = %d AND id_usuario = %d",
            $idCampo, $idUserio);
        if ($result) {
            $usuario_campos_preferencia = db_fetch_object($result);
            $valores_preferencia[$idCampo][] = $usuario_campos_preferencia;
        }
    }
}
```



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

```
foreach ($valores_preferencia as $idCampo => $valor_preferencia) {
    $i = 0;
    $count = count($valor_preferencia);
    for ($i; $i < $count - 1; ++$i) {
        for ($j = $i + 1; $j < $count; ++$j) {
            $valor_i = $valor_preferencia[$i]->valor_maximo_usuario;
            $valor_j = $valor_preferencia[$j]->valor_maximo_usuario;
            if ($valor_j > $valor_i) {
                $temp = $valor_preferencia[$i];
                $valor_preferencia[$i] = $valor_preferencia[$j];
                $valor_preferencia[$j] = $temp;

                $correspondiente = $encontrado[$idCampo];
                $temp2 = $correspondiente[$i];
                $correspondiente[$i] = $correspondiente[$j];
                $correspondiente[$j] = $temp2;
                $encontrado[$idCampo] = $correspondiente;
            }
        }
    }
    $valores_preferencia[$idCampo] = $valor_preferencia;
}
```



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

```
$preferencia_general = array();

$mayores = array();
foreach ($valores_preferencia as $idCampo => $valor_preferencia) {
    $maximo_valor_campo = $valor_preferencia[0];
    $maximo_valor_campo->id_campo = $idCampo;
    $maximo_valor_campo->valor = $encontrado[$idCampo][0]->valor;
    $mayores[] = $maximo_valor_campo;
}

$count = count($mayores);
for ($i = 0; $i < $count - 1; ++$i) {
    for ($j = $i + 1; $j < $count; ++$j) {
        $mayores_i = $mayores[$i]->valor_maximo_usuario;
        $mayores_j = $mayores[$j]->valor_maximo_usuario;
        if ($mayores_j > $mayores_i) {
            $temp = $mayores[$i];
            $mayores[$i] = $mayores[$j];
            $mayores[$j] = $temp;
        }
    }
}

return $mayores;
}
return;
```

Figura 7 Funcionalidad getContenidoAux.Fuente: Elaboración propia.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Funcionalidad getContenidoPersistente.

Es la funcionalidad encargada de retornar los contenidos que cumplen con los campos que se encuentran dentro de las preferencias de los usuarios.

```
function getContenidoPersistente($idTipologia, &$preferencias, $cant, $idUserio, &$recomendaciones) {
    static $predominantes = array();

    if (count($preferencias) === 0) {
        return;
    }
    else {
        $query = "SELECT * FROM (multimedia_campos) mc WHERE mc.id_campo = %d";
        $result = db_query($query, $preferencias[0]->id_campo);

        if ($result) {
            while ($multimedia = db_fetch_object($result)) {
                if (!isset($predominantes[$multimedia->id_multimedia])) {
                    $predominantes[] = $multimedia;
                }
            }

            $i = 1;
            $count = count($preferencias);
            $all = FALSE;
            foreach ($predominantes as $id_multimedia => $predomina) {
                $find = array();
                $not = FALSE;
                for (; $i < $count; ++$i) {
                    $campo = $preferencias[$i];
                    $query = "SELECT * FROM (multimedia_campos) mc WHERE
                        mc.id_multimedia = %d AND mc.id_campo = %d AND mc.valor = '%s'";
                    $result = db_query($query, $id_multimedia, $campo->id_campo, $campo->valor);
                    if (!$result) {
                        $not = TRUE;
                        break;
                    }
                }

                if (!$not && !inHistorial($id_multimedia, $idUserio)) {
                    $recomendaciones[] = $predomina;
                    if (count($recomendaciones) === $cant) {
                        return;
                    }
                }
            }

            unset($preferencias[$count - 1]);
            return getContenidoPersistente($idTipologia, $preferencias, $cant, $idUserio, $recomendaciones);
        }
    }
}
```

Figura 8 Funcionalidad getContenidoPersistente. Fuente: Elaboración propia.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Funcionalidad inHistorial.

Es la funcionalidad encargada de comprobar si un archivo multimedia se encuentra en el historial de un usuario para no mostrarle al mismo contenidos repetidos.

```
function inHistorial($id_multimedia, $idUsuario) {
    $query = "SELECT * FROM {historial} WHERE id_multimedia = %d AND id_usuario = %d";
    $result = db_query($query, $id_multimedia, $idUsuario);
    if ($result) {
        return TRUE;
    }
    return FALSE;
}
```

Figura 9 Funcionalidad inHistorial. Fuente: Elaboración propia.

Funcionalidad getContenidoAleatorio.

Es la funcionalidad encargada de mostrarles a los usuarios contenidos aleatorios en caso que no se encuentre ninguno que sea de su interés o que el mismo no cuente con un historial en la base de datos.

```
function getContenidoAleatorio($cant) {
    $query = "SELECT * FROM {multimedia} ORDER BY id_multimedia DESC";
    $result = db_query($query);
    if ($result) {
        $multimedias = array();
        while ($multimedia = db_fetch_object($result)) {
            $multimedias[] = $multimedia;
        }

        $contenido = array();
        $i = 0;
        $total = count($multimedias);
        for (; $i < $cant; ++$i) {
            $rand = rand(0, $total);
            $contenido[] = $multimedias[$rand];
        }
        return $contenido;
    }
}
```

Figura 10 Funcionalidad getContenidoAleatorio. Fuente: Elaboración propia.

3.3 Validación del algoritmo de personalización a través de un prototipo funcional.

Una de las características típicas del desarrollo de software es la realización de controles periódicos. Estos controles pretenden realizar una evaluación de la calidad de los productos



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

generados para detectar posibles defectos. Sin embargo, todo sistema o aplicación, independientemente de estas revisiones, debe ser probado mediante su ejecución controlada antes de ser entregado al cliente. Estas ejecuciones o ensayos de funcionamiento, posteriores a la terminación del código del software, se denominan pruebas.

El desarrollo y uso de un prototipo funcional como vía para realizar pruebas al software puede ser un camino muy eficaz para identificar y aclarar los requerimientos que deben satisfacer una aplicación, así como para verificar la factibilidad del diseño de un sistema. Las razones para el uso de prototipos funcionales son resultado de la necesidad de diseñar y desarrollar sistemas de información con rapidez, eficiencia y eficacia.

Para analizar los resultados obtenidos luego de implementar el algoritmo propuesto se creó un prototipo funcional para examinar que el mismo efectuara correctamente el proceso de personalización de la interfaz de la plataforma. En la construcción del prototipo funcional fue utilizado el Sistema Gestor de Contenido: Drupal, este gestor es una herramienta libre, realizado sobre la licencia GNU/GPL, además soporta PostgreSQL como gestor de base datos y el Apache como servidor web los cuales son también tecnologías libres. Posee diferentes módulos y una arquitectura que permite añadir nuevas funcionalidades lo cual hace que sea potente y flexible. El lenguaje de programación que usa es PHP, el cual es rápido y soporta varios servidores.

En la figura se muestra cómo funciona el algoritmo una vez que interactúa con el prototipo funcional una vez registrado el usuario en la aplicación.

The screenshot displays the VIDEOWEB website interface. At the top, there is a navigation bar with the site logo and a banner for 'THE WARRIOR WAY'. Below the navigation bar, there are three main sections:

- Inicio de sesión:** A login form with fields for 'Usuario: *' (containing 'usuario1') and 'Contraseña: *' (masked with dots). Below the fields are buttons for 'Iniciar sesión' and a link for 'Solicitar una nueva contraseña'.
- Te recomendamos:** A section featuring a movie poster for 'Repomen' with the title 'Repomen' displayed below it.
- Avisos:** A section with the title 'Avisos' and a link for 'Escuela de Verano 2011: Primer llamado', dated '06/16/2011 - 03:40'.

Figura 11 Muestra sobre los resultados luego de la personalización. Fuente: Elaboración propia.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

En la figura se muestra como el prototipo funcional permite seleccionar la cantidad de recomendaciones que el usuario desea, a partir de los contenidos que figuran dentro de sus gustos y preferencias.

Figura 12 Muestra sobre la selección de la cantidad de recomendaciones que desea el usuario. Fuente: Elaboración propia.

El prototipo funcional brinda la posibilidad de agregar nuevas tipologías, nuevos contenidos multimedia, así como los campos asociados a la misma, como se muestra a continuación:

Tipología	Operaciones		
Película	Editar	Eliminar	Campos
Documentales	Editar	Eliminar	Campos

Figura 13 Agregar nueva tipología. Fuente: Elaboración propia.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Multimedias

[Lista](#) [Agregar multimedia](#) [Tipologías](#) [Campos](#)

▼ **Datos generales**

Título de la multimedia:

El título de la multimedia, Ej. *La fuerza de la verdad*, *El Gladiador*

Tipología de la multimedia:

Descripción

▼ **Archivos relacionados**

Imagen de la multimedia:

Un imagen de multimedia representa por ejemplo el *Poster de una película o una serie*

Archivo de multimedia:

El video de la multimedia

▼ **Otros datos**

Descripción:

Figura 14 Agregar nueva multimedia. Fuente: Elaboración propia.

Multimedias

[Lista](#) [Agregar multimedia](#) [Tipologías](#) [Campos](#)

#	Nombre	Operaciones	
1	Genero	Editar	Eliminar
2	Pais	Editar	Eliminar
3	Director	Editar	Eliminar
4	Anno	Editar	Eliminar
5	Actor	Editar	Eliminar

Nuevo campo: *

Figura 15 Agregar nuevos campos de la tipología. Fuente: Elaboración propia.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

3.4 Resultados del algoritmo desarrollado para personalizar la interfaz de la plataforma VideoWeb.

- La obtención de un algoritmo basado en sistemas de recomendación que explote los datos brindados por los patrones de comportamiento y preferencias de los usuarios.
- La obtención de un algoritmo capaz de brindar una interfaz personalizada pertinente a las necesidades de cada usuario que interactúa con la plataforma.
- La obtención de un prototipo funcional capaz de validar el correcto funcionamiento del algoritmo para la personalización de la interfaz de la plataforma VideoWeb.



CAPÍTULO 3. “Desarrollo del algoritmo y análisis de los resultados”

Conclusiones del capítulo 3

En este capítulo se implementó el algoritmo para personalizar la interfaz de la plataforma VideoWeb, también se creó un prototipo funcional para realizar las pruebas al mismo, el que permitió comprobar que el algoritmo cumplía con el objetivo trazado en la investigación, lo que demostró que mientras más información se pueda obtener sobre las preferencias de los usuarios que interactúan con la plataforma, mejor serán las recomendaciones arrojadas por el algoritmo a los clientes.



Conclusiones Generales

Durante el desarrollo de la investigación, el cumplimiento de las tareas y objetivos propuestos se llegó a las siguientes conclusiones:

- Se presentó un enfoque general acerca del estado del arte de las técnicas de extracción de la información y el proceso de personalización con el fin de personalizar los servicios brindados por la Plataforma VideoWeb.
- Para implementar el algoritmo propuesto se utilizó una composición de técnicas de extracción de información que resuelven el problema de la personalización de la interfaz, ya que ambas técnicas brindaron la información pertinente para desarrollar correctamente el algoritmo.
- Para la prueba del algoritmo se elaboró un prototipo funcional como vía para identificar y aclarar los requerimientos que debe cumplir el algoritmo propuesto.
- El algoritmo propuesto permite la personalización de la interfaz de la plataforma a partir de las preferencias de los usuarios, siendo más efectivo cuando se obtiene mayor información sobre las necesidades de los clientes que interactúan con la plataforma.
- A partir de los resultados obtenidos con el desarrollo del algoritmo propuesto se comprobó la validez del trabajo realizado, brindándole a los usuarios la información adaptada a sus gustos y preferencias.



Recomendaciones

El algoritmo realizado no se considera un producto terminado, si no que se le pueden agregar nuevas funcionalidades que lo complementen para que el mismo en un futuro pueda brindar servicios más personalizados a cualquiera entidad que lo necesite, para ello se recomienda:

- Continuar con la investigación a partir de otros fundamentos y técnicas de extracción de información que permitan obtener contenidos que generen preferencias de usuarios en la plataforma VideoWeb, para hacer de este sistema un producto más completo, lo que permitirá una gran oportunidad en el mercado, para las entidades que utilicen la aplicación.
- Utilizar el algoritmo en distintos proyectos de la universidad que brinden servicios a los usuarios y que requieran de una interfaz personalizada.



Referencias Citadas

Definición. 2008. Definición.org. *Definición de Lenguaje de Programación.* [Online] Marzo 25, 2008. [Cited: Noviembre 11, 2010.] <http://www.definicion.org/lenguaje-de-programacion>.

Falgueras, Benet Campderrich. 2003. *Ingeniería de Software.* s.l. : Editorial UOC, 2003.

Fcom. 2008. Fcom. *Fcom.* [Online] 2008. [Cited: noviembre 15, 2010.] <http://fcom.us.es/fcomblogs/nuevafcom/2008/09/08/los-diez-principios-clave-de-la-personalizacion/>.

Jimmy Wales. 2001. Wikipedia. [Online] 2001. [Cited: octubre 18, 2010.] http://es.wikipedia.org/wiki/Personalización_de_contenidos.

Kernelerror. 2009. [Online] 2009. <http://kernelerror.net/programacion/php/arquitectura-3-capas/>.

—. [Online] [Cited: octubre 13, 2010.] <http://kernelerror.net/programacion/php/arquitectura-3-capas/>.

—. **2009.** *Arquitectura 3 Capas.* *Arquitectura 3 Capas.* [Online] 2009. <http://kernelerror.net/programacion/php/arquitectura-3-capas/>.

—. **2009.** *Arquitectura 3 Capas.* [Online] 26, 2009. [Cited: octubre 21, 2010.] <http://kernelerror.net/programacion/php/arquitectura-3-capas/>.

Mario Claypool. 2001. *Implicit Interest Indicators, Proceedings.* [Online] 2001.

Morfeo Project. 2009. Morfeo Project. [Online] Morfeo-EzWeb, 2009. [Cited: octubre 18, 2010.] http://forge.morfeo-project.org/wiki/index.php/D4.1.1_M%C3%A9todos_y_t%C3%A9cnicas_para_extracci%C3%B3n_de_informaci%C3%B3n_sobre_el_comportamiento_del_usuario_en_distintos_entornos.

MSc. Eleazar Botta-Ferret, Lic. Jania E. 2007. *Minería de textos: una herramienta útil para mejorar la gestión del bibliotecario en el entorno digital.* [Online] julio 31, 2007. [Cited: octubre 28, 2010.] http://bvs.sld.cu/revistas/aci/vol16_4_07/aci051007.html.

Periodismo y Web 2.0. 2006. *Periodismo y Web 2.0.* *Periodismo y Web 2.0.* [Online] 2006. <http://ermelindamaglione.wordpress.com/2009/12/01/cliente-servidor/>.



Pressman, Roger S. 1988. *Ingeniería de Software*. s.l. : Mc Graw-Hill., 1988.

SAS. 2010. The Power To Know. [Online] SAS Argentina Educación, Septiembre 14, 2010. [Cited: octubre 28, 2010.] http://www.sas.com/offices/latinamerica/chile/noticias/notas_prensa/analisis_texto.html.

Sourceforge. 2005. [Online] 2005. [Cited: noviembre 22, 2010.] <http://issi.dsic.upv.es/publications/archives/f-1069167248521/actas.pdf>.

sourceforge.net. 2005. sourceforge.net. *sourceforge.net*. [Online] 2005. [Cited: mayo 11, 2011.] <http://oness.sourceforge.net/proyecto/html/ch03s02.html>.

Sunsite. 2004. Sunsite. *Sunsite*. [Online] 2004. http://sunsite.dcc.uchile.cl/java/docs/JavaTut/Apendice/arq_mvc.html.

—. tutorial de Java. Arquitectura MVC. [Online] [Cited: diciembre 1, 2010.] http://sunsite.dcc.uchile.cl/java/docs/JavaTut/Apendice/arq_mvc.html.

World.com. 2004. Administracion III de Comercializacion. *Administracion III de Comercializacion*. [Online] 2004. [Cited: noviembre 15, 2010.] <http://www.adm3.unlp.edu.ar/regular/material/tecnicasdepersonalizacion.pdf>.

W, Kim. 2002. Personalization: Definition, status, and challenges ahead. Journal. 2002.

YLL, Shelly. 2003. *C++ Guía de Autoenseñanza*. . 2003.

Yunior Ramírez Cruz . 2008. Centro de Estudios de Reconocimiento de Patrones y Minería de Datos. [Online] septiembre 2008. [Cited: octubre 2010, 27.] http://www.cerpamid.co.cu/sitio/files/tesis_yunior.pdf.

Yunior Ramírez Cruz. 2008. Centro de Estudios de Reconocimiento de Patrones y Minería de Datos. [Online] diciembre 23, 2008. [Cited: octubre 27, 2010.] <http://innovacion.ciget.lastunas.cu/index.php/innovacion/article/viewFile/39/35>.



Referencias Bibliográficas

Abreu, Angel Dayán. 2009. Desarrollo de la Plataforma VideoWeb. 2009.

IWorld.com. Técnicas y modelos de personalización de sitios. Sciences de l' Evolution, 2003. Sciences de l' Evolution, 2003. F-34095 Montpellier cdex 05.

Falgueras, Benet Campderrich. 2003. Ingeniería de Software. s.l. : Editorial UOC, 2003.

Gonzalez, Antonio. 2009. zarza.usal.es. [Online] 2009. [http://zarza.usal.es/fgarcia/doctorado/iweb/05-07/Trabajos/MineriaWeb %20y %20/Personalizacion.pdf](http://zarza.usal.es/fgarcia/doctorado/iweb/05-07/Trabajos/MineriaWeb%20y%20Personalizacion.pdf).

Gallego Vázquez, José Antonio en: Guía Práctica para usuarios. 2003 . *Desarrollo Web con PHP y MySQL*. Madrid : Anaya, 2003 . 84-415-1525-5.

Molina, Mería Pinto. 2004. Búsqueda y recuperación de Información. [En línea] 2004. [Citado el: 23 de noviembre de 2010.] http://www.mariapinto.es/e-coms/recu_infor.htm.

Mitecnologico. Concepto de Sistema de Información. [Online] [Cited: noviembre 23, 2010.] <http://www.mitecnologico.com/Main/ConceptoSistemaInformacion>.

Pressman, Roger S. 1988. Ingeniería de Software. s.l. : Mc Graw-Hill., 1988.

Sánchez Enriquez, Hyder Ysaías. 2008. Aplicación en Minería de Datos. Web Mining. 2008.

Software, Colectivo de Autores Asignatura Ingeniería de. 2011. Entorno Virtual de Aprendizaje. [Online] 2011. [Cited: 2 15, 2011.] <http://eva.uci.cu/mod/resource/view.php?id=35381>.

Y. Ramírez Cruz, H. Anaya Sánchez, M. R. Álvarez Silva. 2008. RECONOCEDOR DE NOMBRES DE ENTIDADES PARA EL ESPAÑOL BASADO. [En línea] 23 de diciembre de 2008. [Citado el: 27 de octubre de 2010.] <http://innovacion.ciget.lastunas.cu/index.php/innovacion/article/viewFile/39/35>.

Yunior Ramírez Cruz. 2008. Centro de Estudios de Reconocimiento de Patrones y Minería de Datos. [Online] diciembre 23, 2008. [Cited: octubre 27, 2010.] <http://innovacion.ciget.lastunas.cu/index.php/innovacion/article/viewFile/39/35>.



Glosario de Términos

HTTP: Protocolo de transferencia de hipertexto (en inglés *HyperText Transfer Protocol*) es el método más común de intercambio de información en la World Wide Web, el método mediante el cual se transfieren las páginas web a un ordenador.

Interfaz: Zona de contacto o conexión entre dos componentes de hardware; entre dos aplicaciones, o entre un usuario y una aplicación. Apariencia externa de una aplicación informática.

Internet: Conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.

Multimedia: Elemento que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información.

Software: Conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadoras, es decir, la parte intangible o lógica de una computadora.

Tipología: término utilizado en el proyecto Plataforma VideoWeb para definir grupos de archivos multimedia que son de un mismo tipo.

URL: Son las siglas de Localizador de Recurso Uniforme (en inglés *Uniform Resource Locator*), la dirección global de documentos y de otros recursos en la World Wide Web.