

Universidad de las Ciencias Informáticas

Facultad 1



**Título: Herramienta educativa para la selección de rasgos
utilizando teoría de testores como apoyo a la enseñanza de la
asignatura optativa Reconocimiento de Patrones.**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Silvino de Deus Nduwa Gomes

Tutores: MsC. Maidelis Milanés Luque

Ing. Ilsen León Herrera

La Habana, junio de 2021

“Año 62 de la Revolución”

FRASE

(insertar imagen)

DECLARACIÓN JURADA DE AUTORÍA

Declaro por este medio que yo Silvino de Deus Nduwa Gomes, ser el autor principal de la presente tesis que tiene por título: “Herramienta educativa para la selección de rasgos utilizando la teoría de testores como apoyo a la enseñanza de la asignatura optativa Reconocimiento de Patrones”, y reconozco a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales de la misma, con carácter exclusivo. Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los ____ días del mes de _____ del año 2021.

Silvino de Deus Nduwa Gomes

Firma del Autor

MsC. Maidelis Milanés Luque

Firma del Tutor

Ing. Ilsen León Herrera

Firma del Tutor

AGRADECIMIENTOS

DEDICATORIA

RESUMEN

Dentro de la actual renovación de las metodologías docentes, las universidades están impulsando el uso de herramientas educativas para la enseñanza, ya que permite flexibilizar e individualizar la educación. El objetivo del presente trabajo es crear una Herramienta Educativa para el aprendizaje de pasos para la selección de rasgos utilizando teoría de testores como apoyo a la enseñanza de la asignatura optativa Reconocimiento de Patrones, con la cual se facilitan diferentes demostraciones interactivas y recursos virtuales intencionados al desarrollo de dicha habilidad con el objetivo de que los estudiantes alcancen el entendimiento respecto al tema a mejor nivel posible como también el ahorro del tiempo.

El Reconocimiento Lógico Combinatorio de Patrones, es uno de los enfoques de Reconocimiento de Patrones, resalta por permitir el trabajo simultáneo con variables cuantitativas y cualitativas. Los problemas de selección de rasgos pretenden identificar las características más relevantes que influyen en un fenómeno, en el Reconocimiento Lógico Combinatorio de Patrones se resuelve apoyándose en el peso informacional de los rasgos, comúnmente apoyados por la Teoría de Testores (Testores Típicos).

Todos los recursos que se incluyen en el software tienen un diseño tecnológico. Desde el punto de vista didáctico se tiene en cuenta el desarrollo de las habilidades de rápida adaptación y comprensión a lo máximo posible en la práctica de los estudiantes, según sus necesidades para desempeñar mejor rendimiento en el tema. Desde el punto de vista tecnológico se elabora una aplicación Web Con PHP como lenguaje de programación en el lado del servidor. En el módulo principal el estudiante podrá descargar un archivo de base de datos en el formato CSV para que la aplicación posteriormente pueda demostrar los pasos siguientes del proceso de la selección de los rasgos de la misma.

PALABRAS CLAVE: Herramienta educativa; Reconocimiento Lógico Combinatorio de Patrones; selección de rasgos; Testores Típicos; algoritmos de escala exterior.

RESUMEN

KEYWORDS: .

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

La selección de rasgos es uno de los problemas que se estudia en el área del reconocimiento de patrones, con el objetivo de identificar en un problema aquellas variables que son imprescindibles para la solución del mismo. Existen disímiles maneras en diferentes áreas del conocimiento para identificar los rasgos significativos en los problemas, una de ellas es la teoría de testores, relacionada al área del reconocimiento lógico combinatorio de patrones. Como definición los testores son un conjunto de rasgos que permiten identificar a objetos de diferentes clases en un problema de clasificación.

La selección de variables puede constituir un problema en sí mismo, esto es, nos interesa conocer la influencia de los rasgos en el fenómeno o conjunto de objetos que estamos estudiando. Existen situaciones prácticas en las cuales el interés se centra en las descripciones de los objetos, en la importancia de las variables para describir el problema, en su poder descriptivo. A este problema Fukunaga [1] lo denomina “problema de selección de variables para la representación”.

La otra variante, denominada por Fukunaga [1] en el texto citado como “problema de selección de variables para la clasificación”, es el caso en que nos interesan dos cuestiones fundamentales, a saber, ¿cuáles de las muchas variables que pudiéramos considerar para el estudio de un fenómeno o conjunto de objetos son las que no podemos pasar por alto? y una vez que nos quedamos con un conjunto de éstas ¿son igualmente relevantes para el estudio?, ¿cuál es la relevancia de cada una de ellas?

Las Herramienta educativas son programas educativos didácticos que son diseñados con el fin de apoyar la labor de los profesores en el proceso de enseñanza- aprendizaje; las herramientas educativas están destinadas a la enseñanza y el aprendizaje autónomo y permite el desarrollo de ciertas habilidades cognitivas.

Dentro del Plan de estudio de la carrera Ingeniería en Ciencias Informáticas, se encuentra como asignatura optativa Reconocimiento de Patrones, perteneciente a la disciplina de Inteligencia Computacional. Dentro del contenido impartido se encuentra la selección de rasgos utilizando la

teoría de testores. Aunque el proceso es bastante sencillo, este se complejiza debido a la cantidad de cálculo que es necesario realizar y la longitud de las matrices que se generan. Lo cual dificulta que los estudiantes puedan visualizar este contenido en bases de datos internacionales creadas con fines experimentales para resolver problemas que son propios de la profesión de ingenieros relacionados con el desarrollo de software.

Las herramientas que se utilizan como apoyo a la enseñanza de la inteligencia artificial son : MATLAB (abreviatura de MATrix LABoratory, «laboratorio de matrices») es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M), muy utilizado por las universidades a nivel mundial, tiene un paquete destinado a el entrenamiento de distintos modelos de redes neuronales artificiales. Weka (Waikato Environment for Knowledge Analysis, en español «entorno para análisis del conocimiento de la Universidad de Waikato») es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato. Weka es software libre distribuido bajo la licencia GNU-GPL. A pesar de que ambas herramientas poseen un alto valor y amplio espectro en el uso de algoritmos de selección de rasgos, no incluyen lo relacionado con la teoría de testores.

Teniendo en cuenta lo anteriormente planteado se ha identificado el siguiente **problema a resolver**: La visualización paso a paso de la selección de rasgos utilizando la teoría de testores en el tema I de la asignatura optativa Reconocimiento de Patrones de la carrera de Ingeniería en Ciencias Informáticas.

Objeto de Estudio: Selección de rasgos.

Objetivo General: Desarrollar una aplicación informática para la selección de rasgos utilizando la teoría de testores, como apoyo al Proceso de Enseñanza y Aprendizaje (PEA) en la asignatura optativa Reconocimiento de Patrones.

Campo de acción: Selección de rasgos utilizando la teoría de testores.

Tareas a cumplir por los estudiantes:

- Caracterización del problema de selección de rasgos en el reconocimiento de Patrones.
- Descripción de la teoría de testores como alternativa para la selección de rasgos en problemas de reconocimiento de patrones.
- Identificación de diferentes herramientas informáticas que sirvan como apoyo a la selección de rasgos utilizando la teoría de testores.
- Implementación de la herramienta en función de los elementos estudiados.
- Descripción de futuros desarrollos sobre la herramienta implementada.

Posibles resultados:

Una aplicación informática para la selección de rasgos utilizando la teoría de testores, que apoye al proceso docente educativo de la asignatura optativa que se imparte en la carrera.

Durante la investigación y desarrollo del presente trabajo de diploma se emplearon los siguientes **métodos científicos:**

Métodos Teóricos:

- **Analítico-Sintético:** Este método permite examinar la bibliografía correspondiente a los contenidos de las Herramientas Educativas (HE) y de la selección de rasgos, identificando elementos necesarios para dar solución al problema planteado.
- **Análisis histórico-lógico:** Este método es útil para el estudio crítico de las investigaciones anteriores, de su evolución y en el uso de estos como puntos de referencia y comparación de los resultados alcanzados. Este método permite entender la evolución y el surgimiento de las herramientas educativas, así como las temáticas relacionadas con su implementación.
- **Modelación:** permite la modelación de las funcionalidades del módulo, facilita la representación explícita de la solución propuesta a través de la modelación de la herramienta Educativa, así como de los referentes teóricos extraídos de las fuentes bibliográficas consultadas.

Métodos Empíricos:

- **Observación:** Es útil para constatar las deficiencias existentes en las herramientas utilizadas, así como investigar sobre el funcionamiento, ventajas y desventajas de otras aplicaciones educativas que sirven para el modelado del tutor y dominio.

El presente documento está estructurado en tres capítulos:

Capítulo 1: Fundamentación teórica: este capítulo contiene la base teórica para entender el problema planteado, se realiza una valoración de la importancia de las Herramientas Educativas en la enseñanza para la formación de futuros profesionales. Se argumentan sobre conceptos y pasos para selección de rasgos utilizando la teoría de testores, como también se argumenta el uso de las TIC en la educación caracterizando los Objetos de Aprendizaje como recursos de apoyo al Proceso de Enseñanza y Aprendizaje (PEA) de la asignatura optativa reconocimiento de patrones. También se realiza una descripción de las principales herramientas y tecnologías a utilizar en el desarrollo de la investigación, así como la metodología de desarrollo que guía el proceso de creación del software.

Capítulo 2: Propuesta de solución: en este capítulo se presentan las fases de planificación y diseño definidas por la metodología Proceso Unificado Ágil (AUP, por sus siglas en inglés) variación UCI. Se realiza la propuesta de solución y se planifica el proceso de desarrollo de software. Se definen, entre otros, las descripciones de los casos de usos, los requisitos funcionales y no funcionales y la arquitectura del sistema.

Capítulo 3: Diseño, Implementación y prueba. Se definen los estándares de codificación a utilizar y se realiza el diagrama de componente y de despliegue de la solución. Además, se realizan pruebas para verificar la calidad del módulo del proceso de enseñanza-aprendizaje para la selección de rasgos, se muestran los resultados y las interfaces principales.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se establecen conceptos fundamentales para la comprensión de la investigación. Se realiza un análisis del estado del arte de las herramientas educativas que presentan características similares al módulo que será desarrollado. Se estudian conceptos referidos al Reconocimiento de Patrones desde la perspectiva de varios autores. Además, se realizan una descripción de los algoritmos de escala exterior para el cálculo de todos los testores y se analizan algunas de las herramientas para el RLCP que resuelvan problemas de selección de rasgos. Se realiza una valoración sobre el uso de las tecnologías en el proceso de enseñanza-aprendizaje, así como una breve descripción de las características y funcionalidades del módulo. Finalmente se caracterizan las herramientas, lenguajes y metodologías consideradas para dar solución al problema de investigación.

Herramientas Educativas

La educación se concibe desde una amplia gama de perspectivas. Algunos la perciben solo como un proceso de enseñanza-aprendizaje; otros, la visualizan como una acción laboral; otros, como un servicio y otros, como un derecho. La enseñanza y el aprendizaje son uno solo; pues no puede hablarse de enseñar sin que alguien aprenda; aunque, sí es posible hablar de lo contrario, es decir, de alguien que aprende, sin que nadie le enseñe. Estos dos procesos (la enseñanza y el aprendizaje) hacen parte del desarrollo humano, de la formación de cada una de las personas; pero cuando se habla de enseñanza-aprendizaje, se está hablando de un proceso establecido dentro de un plan institucional con un propósito formal y estrictamente definido.

Utilizar técnicas, modelos o herramientas educativas que no son acordes con la cultura institucional o con las características de los educandos y de los profesores que comparten dicho proceso es aplicar actividades que son o resultan insulsas a la hora de verificar si se está enseñando y si se está aprendiendo. La didáctica requiere de mucho más que generar espacios para que se reúnan las personas a construir conocimiento acerca de un tema específico; requiere una planeación estratégica de los contextos académicos y, en las actuales condiciones, es esta la base para aprovechar las verdaderas ventajas de las herramientas que ofrecen las TIC, sobre las que se dice

que es mucho más “sencillo” el aprendizaje si se utilizan, que si no; afirmación que exige comprender lo que realmente sucede y se requiere al implementarlas.

Es claro que el avance de la sociedad del conocimiento ha desencadenado un acelerado desarrollo de herramientas informáticas que le permiten al ser humano acceder a la información y los saberes de muchas formas; pero, las herramientas informáticas son solamente eso: herramientas, y en esa medida son una pequeña parte de la transformación de la práctica educativa; pese a ello, en muchos casos son abordadas como la solución al quehacer educativo, desconociendo la realidad de los contextos en su implementación. Para afrontar estos retos, es necesario que los docentes se apropien de las herramientas, a partir de capacitaciones que sustenten sus competencias en el uso de estas en ajuste con los lineamientos de los modelos pedagógicos institucionales, en una sociedad del conocimiento que ha derivado en procesos formativos autónomos, rica en contenidos y que ha facilitado el acceso a la información en cualquier momento y desde cualquier lugar. En tal escenario educativo se revela la importancia del diseño digital que “es la concepción de proyectos de comunicación que utilicen eficazmente todo el potencial de los medios interactivos” (Orihuela & Santos, 2000, p. 37).

DEFINICIÓN: Las herramientas educativas son programas educativos didácticos que son diseñados con el fin de apoyar la labor de los profesores en el proceso de enseñanza- aprendizaje; las **herramientas educativas** están destinadas a la enseñanza y el aprendizaje autónomo y permite el desarrollo de ciertas habilidades cognitivas.

Usualmente están enfocados hacia los maestros, profesores o personas que se dedican a la docencia; así como aquellos que desean aprender y desarrollas nuevas habilidades de manera independiente, mejor conocidas como autodidactas.

Las herramientas constan de ciertas características:

- Facilidad de uso
- Capacidad de motivación
- Relevancia curricular
- Versatilidad
- Enfoque pedagógico
- Orientación

- Evaluación

Reconocimiento de Patrones

El término Reconocimiento de Patrones (RP) como muchos otros de las ciencias modernas recibe diferentes interpretaciones, incluso por los propios científicos y técnicos que se ocupan de su desarrollo.

Según José Ruíz Shulcloper: es una ciencia con un fuerte carácter aplicado e interdisciplinario. Está relacionado con procesos (ingenieriles, físicos, matemáticos y computacionales) de datos que provienen de descripciones de objetos (fotos, hologramas, escrituras, jeroglíficos, símbolos, señales bioeléctricas, acústicas, pacientes, zonas geológicas, etc.) con el propósito de obtener (por medio de dispositivos computacionales y/o seres humanos) información que permita establecer las propiedades de ciertos subconjuntos de objetos y/o las relaciones entre ellos [9].

Según Fukunaga: puede ser considerado como un problema de estimación de las funciones de densidad en un alto espacio dimensional y dividir el espacio en las regiones de categorías o clases [5].

Según Anil K. Jain: es el estudio de cómo las máquinas pueden observando el ambiente aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos [16][17].

En la presente investigación se decide adoptar el concepto de José Ruíz Shulcloper, por reflejar un concepto más amplio del Reconocimiento de Patrones y no limitarse solo a funciones básicas.

Esta variedad de criterios entorno al Reconocimiento de Patrones ha traído aparejado diversas formas (metodologías, enfoques) para resolver problemas sus distintos problemas. Atendiendo a los mecanismos utilizados para representar (modelar) estos problemas se describen cuatro enfoques como se muestra en la figura:

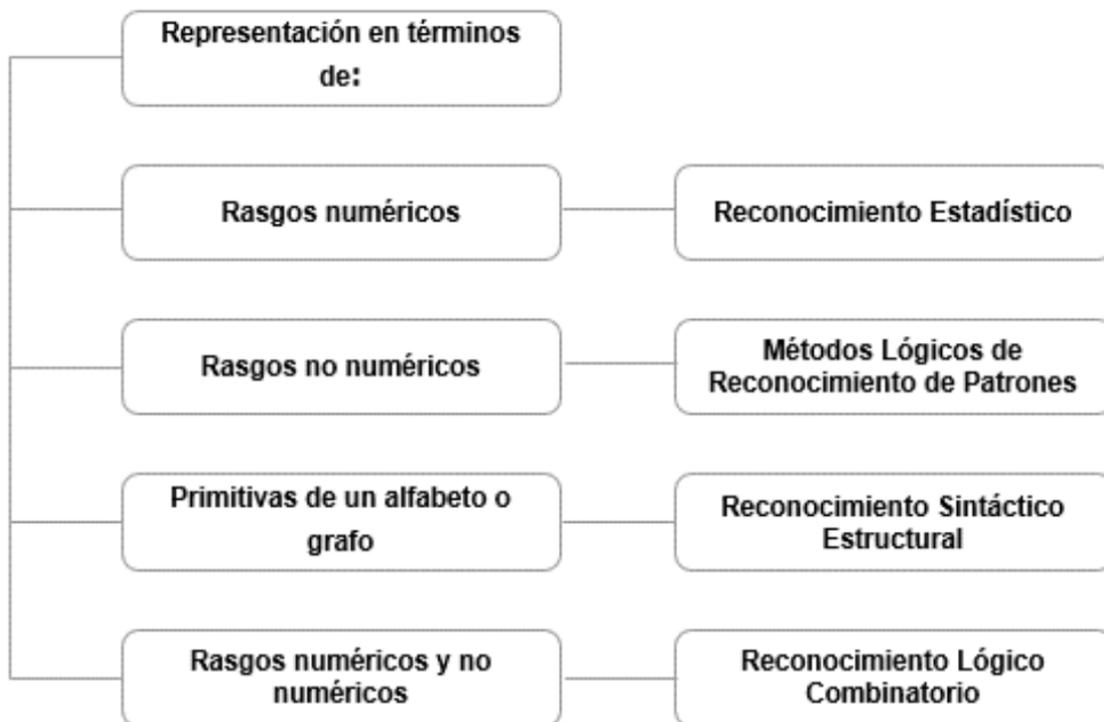


Figura 1 Enfoques del Reconocimiento de Patrones

El presente trabajo de investigación se centra en los presupuestos y teorías del Reconocimiento Lógico Combinatorio de Patrones. El enfoque lógico combinatorio facilita analizar las variables de manera diferenciada lo que permitirá que las mismas puedan ser simultáneamente de naturalezas diferentes (cuantitativas, cualitativas), e incluso permite trabajar con descripciones incompletas de objetos, como es frecuente que ocurra en muchas ramas de la Ciencia, en particular en las zonas poco formalizadas del conocimiento (medicina, geología, sociología), y como aparece en muchos problemas prácticos.

Enfoque Lógico Combinatorio

El enfoque lógico combinatorio al reconocimiento de patrones se nutre de la Lógica Matemática, la Teoría de Testores, la Teoría Clásica de Conjuntos, la Teoría de los Subconjuntos Difusos, la Teoría Combinatoria y la Matemática Discreta en general. Esencialmente los objetos son descritos por medio de una combinación de rasgos de diferente naturaleza de manera que todos (incluidos

los cualitativos) pueden ser procesados por funciones numéricas (Ruiz-Shulcloper, 2002). El primer problema que atiende este enfoque es la selección de variables, y seguidamente puede atender diferentes tipos de problemas de clasificación (Ruiz-Shulcloper, 2002). El problema que trata este trabajo es de selección de variables, lo cual consiste en encontrar los rasgos que inciden en el problema de manera determinante y reducir el número de rasgos en términos de los cuales se deben describir los objetos. Para desarrollar la selección de variables en este enfoque son muy útiles en primera instancia los criterios de comparación para valores de una variable.

Conceptos básicos asociados al Reconocimiento de Patrones

A continuación, se enuncian los principales conceptos relacionados al Reconocimiento Lógico Combinatorio de Patrones para ayudar a una mejor comprensión de la temática abordada en la presente investigación.

Objeto es un concepto con el cual se representa a los elementos sujetos a estudios, (un paciente de una enfermedad, una zona geográfica, un dispositivo eléctrico, un conjunto de personas, una bioseñal, una fotografía, etc.).

Patrón es sinónimo de objeto, solo que en algunas ocasiones es conveniente establecer diferencias entre los objetos, acerca de los cuales se conocen ciertas propiedades, por ejemplo, (la pertenencia a una clase, tipo, etc.) y patrón, de los cuales no se sabe nada acerca de dichas propiedades.

Universo de objetos el conjunto de todos los posibles objetos admisibles para los propósitos de estudio, en cada caso en particular.

Clase es un conjunto de objetos, no solo exclusivamente en el concepto matemático vinculado a la partición de un conjunto (clase de equivalencia), sino también se denomina clases, a aquellos conjuntos que formen un cubrimiento, es decir, se puede tener intersección no vacía.

Rasgo (variable, atributo, característica, propiedad, primitivas de un alfabeto), es un factor a tener en cuenta en el estudio de los objetos dados. Son los rasgos, la vía real para poder trabajar con

los objetos, es decir, se tiene que trabajar con descripciones de los objetos en términos de un conjunto de rasgos.

Dominio de definición es el conjunto de valores admisibles que tiene asociado un rasgo.

A la estructura para almacenar los datos del problema (objetos, rasgos, dominios, clases) se le denominará **Matriz de Entrenamiento (ME)**, se caracteriza por tener m filas (número de objetos) y n columnas (número de rasgos), donde cada fila m_i contienen una descripción de un objeto en términos de n rasgos $\mathfrak{R} = \{X_1, \dots, X_n\}$, estando distribuidos en r clases $\{C_1, \dots, C_r\}$. Cada rasgo X_i toma valores en un conjunto $D_i, i = 1, \dots, n$. [19] [20].

Los mecanismos empleados para determinar qué tan semejantes son los valores de un mismo rasgo en distintos objetos son los **criterios de comparación**.

DEFINICIÓN Sea L_i un conjunto totalmente ordenado (es decir, para cualesquiera dos elementos de L_i siempre podemos determinar cuál de ellos antecede al otro según el orden definido). Sobre M_i , el conjunto de valores admisibles del rasgo x_i , definimos una función $C_i: M_i \times M_i \rightarrow L_i$, que denominaremos criterio de comparación de valores de x_i , tal que:

- A) $C_i(x_i(O), x_i(O)) = \min \{y\}$, siendo $y \in L_i$ (denota el mínimo de L_i) si C_i es un criterio de comparación de *disimilaridad* entre valores de x_i .
- B) y $C_i(x_i(O), x_i(O)) = \max \{y\}$, siendo $y \in L_i$ (denota el máximo de L_i) si C_i es un criterio de comparación de *similaridad* entre valores de x_i , para $i = 1, \dots, n$. [29]

En el RLCP el método empleado para determinar la similitud (o semejanza) entre dos objetos se denomina **función de semejanza entre objetos**. Una función de semejanza Γ se define como una función que se aplica a un par de objetos (O_i, O_j) obteniendo como resultado un valor real que indica qué tan semejantes son dichos objetos. Formalmente se puede definir como $\Gamma: \mathbb{O} \times \mathbb{O} \rightarrow \mathbb{R}^+$. Existen dos tipos de funciones de semejanza: la similitud y la disimilitud. Si es una similitud cumple que mientras más grande sea el valor de $\Gamma(O_i, O_j)$, más semejantes son los objetos O_i y O_j . En cambio, si es una disimilitud mientras más pequeña sea la evaluación de $\Gamma(O_i, O_j)$, más semejantes son los objetos. [30]

Sea una matriz de entrenamiento (ME) que contiene m objetos descritos en términos de n rasgos $\mathfrak{R} = \{X_1, \dots, X_n\}$ distribuidos en r clases $\{C_1, \dots, C_r\}$. Cada rasgo X_i toma valores en un conjunto $D_i, i = 1, \dots, n$. Dado un criterio de comparación de diferencia $\psi_i: D_i \times D_i \rightarrow \{0,1\}$ se asocia a cada X_i ($0 = similar, 1 = disimilar$), para todo posible par de objetos que pertenecen a diferentes clases de ME se crea una **matriz booleana de diferencia** (MD). [24]

Sean p y q dos filas de MD , p es subfila de q si en todas las columnas donde p tiene un 1, q también lo tiene. Se llama **fila básica** p de MD , si ninguna fila en MD es subfila de p . La submatriz de MD que contiene todas las filas básicas (sin repeticiones) se llama **matriz básica** (MB). [24]

Los conceptos antes expuestos, constituyen la base para comprender los problemas fundamentales de los que se ocupa el Reconocimiento de Patrones.

Problemas del Reconocimiento de Patrones

Como se mencionaba inicialmente los problemas del RP son todos aquellos relacionados con la clasificación de objetos y fenómenos y con la determinación de los factores que inciden en los mismos. Así, se consideran cuatro familias de problemas que se denominan respectivamente:

1. Selección de rasgos (o variables) y objetos.
2. Clasificación supervisada (o con aprendizaje).
3. Clasificación no supervisada (o sin aprendizaje).
4. Clasificación parcialmente supervisada (o con aprendizaje parcial).

Clasificación supervisada:

Este es el problema más conocido del RP y con el que más se está relacionado, pues mucha de la actividad humana está vinculada, directa o indirectamente, con procesos de clasificación. Dado un universo de objetos y el conocimiento acerca de la existencia de ciertas clases con características (propiedades) de especial interés y una muestra de objetos que pertenecen a cada una de ellas, el

problema consiste en determinar para cada uno de los objetos no clasificados las relaciones de pertenencia de los mismos con cada una de las clases. [2]

Clasificación no supervisada:

El problema de la clasificación no supervisada (problemas de agrupamiento – clustering, structuralization, en inglés) es una de las etapas más importantes en la mayoría de las ciencias, en particular, las naturales y sociales. La Taxonomía es el primer período de trabajo científico en muchas de las áreas del conocimiento y es una de las herramientas necesarias en muchas de las investigaciones en la actualidad. [2]

El problema principal a resolver es encontrar las relaciones entre los objetos de un universo en términos de sus características (rasgos). Estas relaciones se establecen sobre la base del concepto de analogía (similaridad). Quizás uno de los conceptos más importantes en el RP. [2]

El propósito es juntar (agrupar) los objetos según su analogía (parecido, semejanza, cercanía si se está hablando de un espacio de representación con distancia definida). En este sentido se pueden encontrar dos situaciones diferentes, a saber, el número de grupos (clusters) es conocido previamente o no. [2]

Clasificación parcialmente supervisada:

Esta es quizás la familia de problemas de RP en la que menos estudio se ha realizado. A pesar de la existencia de muchos problemas reales donde aparecen situaciones de este tipo, la clasificación parcialmente supervisada no ha recibido igual atención que las restantes familias de problemas en RP. [2]

El problema consiste en una combinación de los problemas de clasificación anteriormente descritos. En el universo de objetos dado, se conoce la existencia de ciertas clases e incluso se tienen muestras de algunas de ellas, pero no de todas. El problema central es clasificar nuevos objetos en estas circunstancias, en la que no se tienen muestras de todas las clases y en las que incluso pudieran existir clases que se desconocen. [2]

Por ejemplo, se sabe que en una habitación hay un grupo de personas conversando. No se conocen las voces de todas las personas, que se encuentran en la habitación. Se puede incluso no conocer cuántas personas se encuentran en la habitación. El problema pudiera ser determinar qué dijo cada persona durante la conversación. [2]

En el próximo epígrafe se describe detalladamente el problema de selección de rasgos y/o objetos.

Selección de rasgos y/o objetos

La selección de rasgos y/o objetos es uno de los pasos fundamentales en cualquier problema de clasificación debido a que la mayoría de los problemas de RP están basados en la descripción de los objetos en términos de un conjunto de rasgos. [2]

En el primer grupo de problemas, lo que se enfrenta es la determinación del mejor subconjunto de rasgos para la clasificación de nuevos objetos (no clasificados). Esto conlleva la reducción del conjunto de todos los posibles rasgos (reducción de la dimensionalidad) sobre la base de las diferencias que estos rasgos presentan en cuanto a reconocer, clasificar a los nuevos objetos y otros problemas de optimización adicionales del subconjunto de rasgos a emplear, que también se deben tener en cuenta en muchos problemas de la realidad, como la eficiencia computacional entre otros. [2]

Este problema conlleva la determinación de un subconjunto de rasgos que de una mejor manera caracteriza a los objetos de cada una de las clases. Es claro que estos problemas asumen la existencia previa de las clases o en su defecto, la repercusión que estos rasgos tuvieron en la formación de la misma. [2]

De forma general el problema de selección de rasgos y/o objetos consiste en encontrar el subconjunto de rasgos que mejor describe los objetos del dominio; usualmente este subconjunto se encuentra maximizando o minimizando una función objetivo. Se reportan trabajos en la literatura para seleccionar rasgos relevantes con diferentes enfoques y técnicas, tales como: técnicas de optimización de colonias de hormigas, las de computación evolutiva, las basadas en la teoría de los conjuntos aproximados y el enfoque Lógico Combinatorio, problemas como la determinación

de síndromes, factores de riesgos, perfiles de usuarios, características discriminantes de conjuntos de objetos o fenómenos, y muchos más, son instancias de este tipo de problemas en el mundo real. [2] [31]

Este problema se apoya en la selección del peso informacional de los rasgos y lo calcula a partir de la Teoría de Testores. [2] [31]

Peso informal de los rasgos

El peso informacional de un rasgo (o variable), es una medida de su importancia para la clasificación. En su determinación intervienen los valores de cada variable, así como los pesos informacionales de estos [28].

El cálculo del peso informacional de los rasgos se realiza a partir de la Teoría de Testores, con el fin de seleccionar los rasgos que mejor describen a los objetos o los que mejor los clasifican.

Teoría de Testores

La Teoría de Testores es una rama de la Lógica Matemática que surgió en la Unión Soviética a finales de los años sesenta. Su origen está vinculado a la utilización de métodos lógicos matemáticos para la localización de desperfectos en circuitos eléctricos que realizan funciones booleanas. [32]

Los primeros investigadores que desarrollaron estas teorías fueron, I. A. Cheguis y S. V. Yablonskii en lo concerniente a los circuitos lógicos y Y. I. Zhuravliov en la introducción de esta naciente teoría al RP. [32]. Este último aplica el concepto de testor en la solución de un problema de la Geología y presenta los primeros algoritmos para el cálculo de todos los TTs de una matriz. Es entonces que la Teoría de Testores abre toda una rama de investigaciones en la que labora un gran número de especialistas. [32]

Testor (Según la definición extendida de Zhuravliov): El conjunto $\tau = \{X_i, \dots, X_{i_p}\} \subseteq R$ es un testor de una matriz(tabla) T_{nmr} si al eliminar todas las columnas de T excepto las de τ , no aparecen nuevas subdescripciones semejantes en clases diferentes.[Libro de Schulpter]

Testor Típico (TT): Un testor se llama **irreducible (típico)** si al eliminar alguna de sus columnas deja de ser testor.

Peso informacional de los rasgos en función del concepto de Testor

El enfoque basado en la Teoría de Testores se basa sobre la idea de que un testor típico, según Zhuravliov, es un conjunto de columnas (rasgos) de una matriz de entrenamiento para las cuales se cumple que no existen filas iguales en clases diferentes, pero además si eliminamos una cualquiera de estas columnas las restantes perderán esta propiedad [8].

La idea básica consiste en lo siguiente: dada la definición de testor y de testor típico, se tiene que cada objeto se pone en correspondencia con un s-uplo que tiene la propiedad de ser diferente en al menos alguna coordenada con respecto a cualquier otro objeto que pertenezca a una clase diferente. Al pasar de un testor a un testor típico (eliminando rasgos) se llega a una combinación irreducible, donde cada rasgo resulta imprescindible para mantener las diferencias entre las clases. Esta es la propiedad que distingue a los testores típicos. Es natural suponer pues, que, si un rasgo aparece en muchas combinaciones irreducibles, resulta más difícil prescindir de él para describir de manera diferenciante las clases, o sea que él es más diferenciante. Sobre la base de esta idea, Zhuravliov formula su definición de peso informacional de un rasgo como la frecuencia relativa de aparición de ese rasgo en la familia de todos los testores típicos [8][25].

Sea τ el número de testores típicos que tiene una cierta matriz de entrenamiento de un problema de reconocimiento de patrones y sea $\tau(i)$ el número de aquellos testores típicos en los que aparece la columna correspondiente al rasgo x_i . Se dice que el peso informacional (relevancia) de este rasgo viene dado por la magnitud: [25] [8]

$$P(\chi_i) = \frac{\tau(i)}{\tau} \text{ para } i = 1, \dots, n \text{ } x_i \in \mathbb{R}$$

Ecuación 1: Peso informacional del rasgo

Utilizando esta definición se han resuelto una serie de problemas interesantes, sin embargo, la expresión antes propuesta por Zhuravliov para la evaluación del peso informacional de los rasgos, si bien resulta intuitiva, al considerar más importante un rasgo mientras más veces aparece en la familia de todos los testores típicos, no considera las longitudes de los testores que contienen al rasgo en cuestión, de modo que, por ejemplo, si la familia de testores típicos de un problema fuera: [25]

$$\psi^* = \{\{x_5\}, \{x_3, x_1\}, \{x_2, x_3, x_4\}\}$$

, se tiene que:

$$P(x_1) = P(x_2) = P(x_4) = P(x_5) = 0.333 \text{ y } P(x_3) = 0.666$$

Pero resulta natural pensar que el rasgo x_5 es más relevante que x_3, x_4 y x_1 pues aunque sólo aparece en un testor típico no requiere de ningún otro rasgo para lograr diferenciar los objetos que están en clases diferentes, x_1 requiere de otro rasgo y por su parte x_3 y x_4 requieren de otros dos; luego si se denota por $p(x)$ la relevancia del rasgo x , podría esperarse que $p(x_5) > p(x_1) > p(x_2) = p(x_4)$ [25]

Este análisis conduce a concluir que considerar la frecuencia de aparición de un rasgo en la familia de los testores típicos como medida de su importancia, aunque resulta bastante natural a partir de la interpretación de un testor típico como una combinación irreducible de rasgos diferenciadores, en algunos casos esto puede llevar a interpretaciones erróneas [25].

Se asociará a cada rasgo una magnitud, que depende de estas longitudes y que se denotará por: [25]

$$L(x_i) = \frac{\sum_{t \in \psi^*} \frac{1}{t}}{|\psi^*(x)|}$$

Ecuación 2: Peso del rasgo dependiendo de la longitud del Testor

Siendo $\psi^*(x)$ la familia de todos los testores típicos que contienen al rasgo x .

Para el ejemplo anterior se tiene que:

$$L(x_5) = 1, L(x_3) = 0.418, L(x_3) = L(x_4) = 0.333 \text{ y } L(x_5) = 0.5.$$

Estos valores no se corresponden totalmente con lo que se señalaba como deseado con respecto a la relevancia, pues, aunque x_5 aparece con máximo valor, x_1 tiene asociado un valor mayor que el de x_2 y debía esperarse lo contrario.

Entonces como ya se tiene que $P(x)$ toma en cuenta sólo la frecuencia de aparición del rasgo en la familia de testores típicos; $L(x)$ por su parte considera únicamente la longitud de los testores típicos en los que aparece el rasgo.

Se define una nueva medida del peso informacional (relevancia) del rasgo en función de estas dos magnitudes dada por la magnitud:

$$p(x) = \theta(P(x), L(x))$$

Siendo $P(x)$ como en (ecuación 1); $L(x)$ como en (ecuación 2) y θ una función de estas dos magnitudes.

Considerándose la siguiente expresión como ejemplo [25]:

$$p(x) = \alpha P(x) + \beta L(x) \text{ con } \alpha, \beta > 0 \text{ y } \alpha + \beta = 1$$

siendo α y β dos parámetros que ponderan la participación o influencia de $P(x)$ y de $L(x)$ respectivamente en $p(x)$. Esto es la importancia que se concede para la relevancia de la frecuencia de aparición y a la longitud de los testores.

Si se le concede a ambos la misma importancia se considerará $\alpha = \beta = 0.5$. Para estos valores en el ejemplo anterior se tiene:

$$p(x_5) = 0.666, p(x_3) = 0.542, p(x_2) = p(x_4) = 0.333 \text{ y } p(x_1) = 0.416$$

Algoritmos para el cálculo de todos los Testores Típicos

Para el cálculo de los testores típicos se han desarrollado varios algoritmos que por su estrategia de cómputo pueden clasificarse en algoritmos de escala interior y de escala exterior.

En la presente investigación se trabajará con los algoritmos de escala exterior, ya que en la bibliografía [2] los autores toman la misma decisión.

Algoritmos de Escala Exterior

Los algoritmos de escala exterior son aquellos que realizan el cálculo de los TTs generando los elementos del conjunto potencia del conjunto de columnas de la MB en un determinado orden (usualmente utilizando un n-uplo booleano característico), de tal forma que, usando determinados criterios, el algoritmo trata de evitar el análisis de todos los subconjuntos. [29]

En la Figura 2 se muestra un diagrama de flujo de estos algoritmos. De forma general estos algoritmos inician con la matriz básica, posteriormente se generan combinaciones del conjunto potencia en base dos, se comprueba si combinación generada es testor, si es testor se verifica que sea típico, si se cumple esa condición se guarda y se procede a realizar el salto TT, se verifica si se puede hacer otra posible combinación, en caso de poder se genera una nueva y en caso contrario se finaliza. Si la combinación generada del conjunto potencia no es testor, se realiza el salto no testor y procede de la misma forma que el TT después de haber realizado el salto que se define para este.

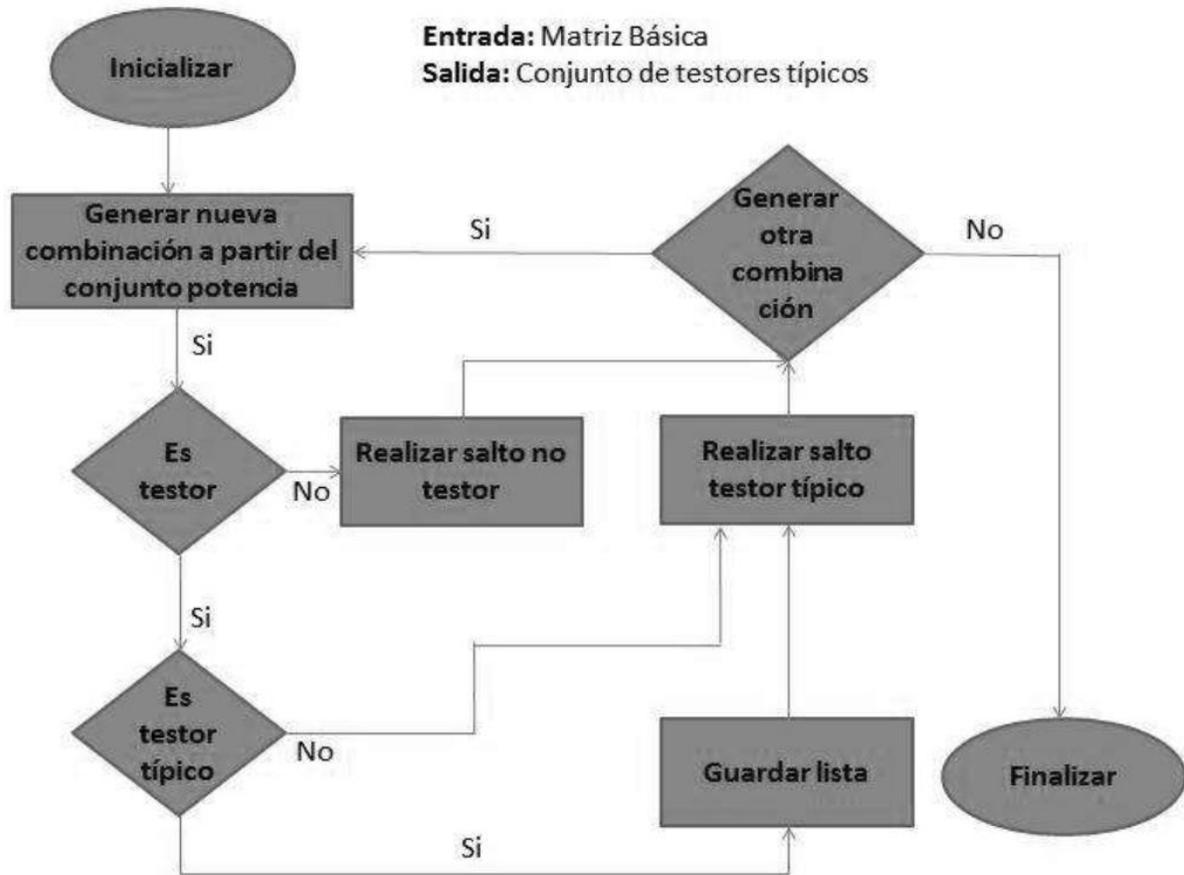


Figura 2 Diagrama de flujo algoritmos de escala exterior

Algoritmo BT

El algoritmo BT está basado en los trabajos de especialistas soviéticos. El orden que caracteriza a este algoritmo es el generado por los números naturales de forma ascendente en su notación binaria mediante un n-uplo booleano. Este constituye un orden total sobre los elementos no nulos del conjunto potencia del conjunto de rasgos de la MA.

El algoritmo comienza entonces por el n-uplo $(0, \dots, 0, 1)$ y procede comprobando si cada vector (n-uplo) generado es un testor o un TT. Según el resultado establecen “saltos” en el conjunto potencia. Dichos saltos se dividen en: saltos no testor y saltos testor, dependiendo si el n-uplo analizado no es testor o sí lo es. Si el n-uplo es TT se saltan todos aquellos n-uplos que son supratestores de este, y se almacena cada TT resultante en una lista (conjunto de todos los TT).

Por otro lado, si el n-uplo es no testor, se observa la fila o las filas que impiden que sea testor, y se escoge aquella que tiene el 1 más a la derecha de la fila que impide que sea testor. [29] [33] [34] [35]

El algoritmo termina cuando llega al n-uplo $(1, \dots, 1, 1)$. Para ello se apoya en caracterizaciones de los conceptos de testor y TT y en proposiciones que definen los saltos cuando el n-uplo analizado es o no un testor. [29] [33] [34] [35]

Algoritmo TB

El algoritmo TB posee iguales ventajas que el BT en cuanto al número de listas a considerar, pero resulta más ventajoso para los casos en que la longitud media de los TT es mayor que $n/2$, donde n es la cantidad de columnas de la matriz básica. Siendo este una mejora del algoritmo anterior. [33]

El TB procederá de manera simétrica, comenzando por la lista unitaria y generando las restantes en el orden antinatural hasta las listas nulas. [33]

Algoritmo BR

El algoritmo BR evita el análisis de un mayor número de subconjuntos irrelevantes y eficientemente verifica la propiedad de testor tomando como ventaja las operaciones a nivel de bits de la computadora. El nombre del método se debe a las operaciones Binarias y a la estrategia Recursiva que éste utiliza. [24]

El método propuesto primeramente reordena las filas y columnas de MB con el objetivo de reducir el espacio de búsqueda de los TT. La fila con el menor número de 1s y el máximo número de 1s en las columnas de MB donde ésta tiene un 1 se coloca como primera fila. El reordenamiento de las columnas permite al algoritmo finalizar cuando el rasgo a ser analizado tenga un cero en la primera fila de MB, también intenta reducir la posibilidad de que los rasgos a ser analizados no sean excluyentes con una lista de rasgos y, por consiguiente, minimiza la longitud de las listas de rasgos que deben ser analizadas. [24]

La idea general del algoritmo BR es primeramente generar listas de rasgos que satisfagan la propiedad de tipicidad y luego, verificar la condición de testor. El método explora el conjunto potencia de los rasgos comenzando por el primer rasgo en la MB y genera una lista de rasgos candidata a ser TT. Una vez que ha sido generada una lista de rasgos candidata, se verifican las propiedades de testor y tipicidad. [24]

Algoritmo LEX

El nombre de LEX se debe al orden lexicográfico en que son comparadas las cadenas de caracteres. La idea del algoritmo es ir construyendo listas de rasgos que posean la propiedad de tipicidad y luego, comprobar si este conjunto de rasgos constituye un testor. Cuando se obtiene un TT se producen saltos. [35]

El algoritmo empieza analizando el primer rasgo de una matriz básica MB. Un nuevo rasgo se puede incorporar a la lista si se cumple que no es excluyente con la lista, es decir, si puede coexistir con los rasgos de la lista para formar un TT y tiene filas típicas con respecto a la lista. Posteriormente, se comprueba si el conjunto de rasgos contenidos en la lista junto con el nuevo rasgo forma un testor, verificando si no existe fila en la MB (tomando en cuenta únicamente esos rasgos) completa de ceros. Si se cumple que es testor, entonces estamos en presencia de un TT y se almacena. [29] [35]

Si se encontró un TT y éste contiene al último rasgo de MB, entonces se saltan todos sus subconjuntos consecutivos. Si no contiene al último rasgo de MB, para saltar todos los supraconjuntos del TT encontrado se elimina el último rasgo de la lista y se analiza si se puede incluir el próximo rasgo de MB. Si el rasgo analizado no se puede incorporar a la lista se prosigue el análisis con el siguiente rasgo de la MB. [29] [35]

Cabe indicar que previamente se realiza un ordenamiento de la MB que consiste en colocar como primera fila aquella que tenga la menor cantidad de unos (ordenar filas), y en esta fila encontrada, modificar las columnas de tal modo que se sitúen todos los unos a la izquierda (ordenar columnas). Si existiera más de una fila con mínima cantidad de unos se puede utilizar el concepto de Entropía

Global para la selección. Así, se escoge aquella fila que tiene la mayor entropía global, o sea, aquella en la que la suma de los unos presentes en las columnas donde esta fila tiene valores unitarios sea mayor. [29] [35]

Algoritmo REC y CER

El REC tiene la particularidad de que trabaja directamente con la matriz de aprendizaje, lo cual lo sitúa en desventaja con relación a los restantes algoritmos por el enorme manejo de información que tiene que realizar. [29]

El CER fue encaminado a resolver este problema y lo hizo introduciendo un nuevo orden en el conjunto potencia de los rasgos. [29]

Justificación de la selección del algoritmo

El algoritmo BT, resulta ser como se observó en las descripciones anteriores uno de los algoritmos básicos de escala exterior, a partir de los cuales se han desarrollado nuevas mejoras. Este a su vez según la bibliografía consultada [36] [37] [38] [39] resulta ser un punto de comparación para el desarrollo de nuevos algoritmos.

En el estudio realizado en [37] sobre los distintos tiempos de ejecución de los algoritmos de escala exterior como se muestra en la tabla se pueden observar con mejores resultados al LEX y el BR.

Tabla 1. Tiempo del funcionamiento de los algoritmos para varias matrices básicas

Juegos de Datos	Clases	MB	LEX	CT-EXT	BR	NTT
Zoo(101x17)	7	14x17	0:0:00:15	0:0:0:718	0:0:00:00	34
Mushroom(8124x22)	2	30x22	0:0:00:16	0:0:0:750	0:0:00:00	292
Chess(3196x36)	2	29x36	0:2:22:16	0:8:01:67	0:0:00:12	4
Dermatology(366x34)	6	1124x34	0:25:45:7	1:43:15:6	0:0:58:22	115 556
Promoter(106x57)	2	2761x57	1:07:27:5	4:24:23:8	0:3:18:51	7 456 943
Random	_2	150x70	0:55:45:3	2:06:30:4	0:4:02:67	44 165 054
Random	_2	100x100	2:22:01:9	>20 horas	0:10:30:1	183 051 234

En una encuesta realizada a un grupo de personas (**ver Anexo 1**), con conocimientos en relación al RLCP, los algoritmos más conocidos son el BT, TB, LEX y BR. Se decide implementar en la presente investigación: El algoritmo BT al ser punto de referencia en investigaciones de estas teorías.

Herramientas Informáticas para la selección de rasgos

Se realizó una búsqueda de las herramientas para la solución de problemas del RP. Estas herramientas serán analizadas tomando en cuenta:

1. El trabajo simultáneo con variables cualitativas y cuantitativas. Para cada tipo de variables se debe contar con diversos criterios de comparación entre dos valores de una misma variable. Se debe permitir para todos los tipos de datos la ausencia de información o valor “?” en su dominio.
2. Dominios en variables descriptivas y manejo intrínseco de los dominios de datos más comunes (enteros, booleanos, reales, etc.), con posibilidad para su extensión atendiendo a las necesidades propias del problema.
3. Criterios de comparación y funciones de analogía entre patrones. Brindar al usuario útiles para la creación de funciones de analogía que permitan la extensión de la herramienta.
4. La extensión con nuevos algoritmos para el procesamiento de la información desarrollados por los usuarios.
5. Permitir comparar los resultados sobre un mismo juego de datos al aplicar diferentes algoritmos.
6. No debe presuponer un tipo especial de espacio de representación para los datos (métrico, vectorial, euclidiano, booleano, etc.). El espacio dependerá en gran medida del problema a modelar y de los datos que sean necesarios utilizar.

Antes de analizar las herramientas encontradas es necesario conocer la definición de Sistema Herramienta. Un Sistema Herramienta se refiere a un complejo de programas orientados a la solución de una familia de problemas de una o más áreas específicas del conocimiento, tal que el usuario sólo requiere conocer el área específica. Si el sistema es aplicable a cualquier área del conocimiento se le denominará Universal. [40]

ACLAPIP

Es un sistema de programas dirigido a la solución de problemas de clasificación, diagnóstico y pronóstico de objetos o fenómenos en los que el valor informativo de cada objeto o estado del fenómeno en estudio es diferente a los demás y esto se desea tener en cuenta. El sistema brinda variantes posibles de aplicaciones del modelo de RP basado en el peso informacional de los objetos acorde a la Teoría de Testores. [43]

ACLAPIP consta de un programa principal y 24 subrutinas. Desde el programa principal se realiza una secuencia de llamadas a subrutinas que realizan las diferentes variantes de cuatro parámetros (Patrón, RSOL, Distancia, Unicidad) esenciales que caracterizan a los 36 algoritmos que efectúa el sistema. [43]

La importancia de ACLAPIP radica fundamentalmente en ser la primera herramienta de su tipo diseñada y construida desde el enfoque Lógico Combinatorio. Se trata en efecto de la primera herramienta de software para cálculo de testores y de las diversas medidas de ellos se desprenden (como la relevancia informacional de objetos y rasgos en un problema de RP). Sin embargo, no se puede decir mucho sobre su flexibilidad pues, en términos prácticos, el usuario no podía modificar la función de semejanza considerada ni el tipo de Testor para los cuales se calcularía la relevancia de los objetos y rasgos. [1] [43]

Ventajas

- Permite indicar la regla de solución que se desea utilizar en la solución del problema.
- Expresa si el problema en cuestión admite como solución la clasificación de un objeto en varias clases simultáneamente o no.

Desventajas

- Ausencia de mecanismos para definir funciones de semejanzas en patrones.

- No posibilita plantear problemas bajo condiciones de comparación diferentes a las definidas por el sistema.
- No presenta mecanismos de comparación entre los resultados obtenidos por diferentes algoritmos aplicados sobre un mismo conjunto de problemas.

En la presente investigación debe tenerse en cuenta muchas de las características mejorables de la herramienta, en función de seleccionar una herramienta con más potencialidades para el RLCP.

PROGNOSIS

Se realizó en el año 1993 con el objetivo de ser un sistema herramienta para la resolución de problemas de RLCP con clases duras. Fue construido tratando de resolver problemas planteados por especialistas en geología, geofísica y sismología, en cuanto a crear una verdadera herramienta de trabajo y de obtención de nuevos conocimientos. Es uno de los pocos sistemas capacitados para abordar el problema de la clasificación con aprendizaje parcial de sus clases, con lo cual se coloca entre los menos que permiten toda la gama de problemas de clasificación y tareas afines. [41]

PROGNOSIS ha sido empleado en el Centro de Investigaciones Sismológicas (CENAI) del CITMA (Ministerio de Ciencia, Técnica y Medio Ambiente), en el Centro de Investigaciones para el Desarrollo del Petróleo (CIDP) entre otras instituciones. Tiene como característica que resuelve los cuatro problemas de RP [41], este sistema utiliza el algoritmo BT para calcular TTs. [34]

Ventajas

- No se presupone un tipo especial de espacio de representación de los datos (métrico, vectorial, euclidiano, booleano, etc.). El espacio dependerá en gran medida del problema a modelar y de los datos que sean necesarios utilizar, sin restricciones en cuanto a tipo.
- Los dominios de definición de las variables pueden ser de diversa naturaleza, cuantitativos o cualitativos; reales, booleanos, k-valentes.

- Ofrece un conjunto de funciones interconstruidas de las cuales el usuario puede seleccionar la que será usada en cada problema.

Desventajas

- Carece de mecanismos para definir nuevas funciones de semejanza.
- No existe forma de que el sistema incorpore algoritmos programados por los usuarios y los pueda aplicar a conjuntos de datos previamente definidos.

PROGNOSIS ha sido aplicado con éxito en proyectos trascendentes de prospección petrolera y análisis geológico [41], pero aún dista de ser adecuado para la generalidad de los investigadores en el RPLC.

WEKA

Está constituido por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación, y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados. Los datos de entrada a la herramienta, sobre los que operarán las técnicas implementadas, deben estar codificados en un formato específico, denominado Attribute-Relation File Format (extensión ".arff"). [44] [45]

La herramienta permite cargar los datos en tres soportes: archivo de texto, acceso a una base de datos y acceso a través de internet sobre una dirección URL de un servidor web. En el caso de los archivos de texto podemos generarlo con cualquier editor de texto, pero al guardarlo debemos modificarle la extensión ".arff". [44] [45]

Ventajas

- Permite declarar nuevos algoritmos y trabaja con ausencia de información.
- Es una herramienta multiplataforma.

Desventajas

- No permite declarar nuevos algoritmos programados desde la perspectiva del Reconocimiento Lógico Combinatorio de Patrones.
- No cuenta con capacidad para permitir el trabajo simultáneo con variables cualitativas y cuantitativas como lo hace el RLCP.

Esta herramienta tiene como principal inconveniente el no estar basada en las teorías del RLCP por lo que no es opción para incorporar algoritmos basados en problemas de selección de rasgos y/o objetos para el cálculo de todos los TTs.

CEPAR

CEPAR (Entorno Cubano para el Reconocimiento Lógico Combinatorio de Patrones) v1.0.0, es un Sistema Herramienta Universal para el RLCP, tiene como objetivo fundamental apoyar en las labores cotidianas de los investigadores, docentes y estudiantes del RLCP, además proveer de una herramienta que pueda ser embebida en desarrollos propios de una investigación. [40]

Desarrollado empleando JAVA como lenguaje de programación, dado sus comodidades como lenguaje multiplataforma, CEPAR solo requiere de la Máquina Virtual de Java (JVM) para poder ser utilizado.

CEPAR plantea un diseño modular, de manera que permite la incorporación de nuevos rasgos, funciones de semejanza, o algoritmos. Cada módulo es independiente y se relaciona con los demás a través de las clases e interfaces definidas en su núcleo (cepar.core), que contiene las características más generales para modelar un problema de RLCP. [40]

De forma nativa CEPAR maneja los tipos de rasgos:

- BooleanFeature: valores booleanos.
- DateFeature: valores de tipo fecha (día-mes-año).

- DoubleFeature: valores reales.
- FloatFeature: valores de punto flotante.
- IntegerFeature: valores enteros.
- StringFeature: valores alfanuméricos.

En la práctica esta cantidad de rasgos resulta minúscula, en comparación con los que pueden aparecer en problemas reales del RLCP, por lo que resalta en CEPAR el permitir la extensión según las interfaces definidas de nuevos rasgos por los usuarios, no limitando la herramienta solo al trabajo con los seis rasgos nativos.

Ventajas

- Para todos los tipos de rasgos soporta la ausencia de información o valor "?".
- La herramienta permite el trabajo con variables cuantitativas y cualitativas (simultáneamente).
- Permite definir criterios de comparación y funciones de analogía entre patrones, de manera que puede extenderse con nuevos algoritmos para el procesamiento de la información desarrollados por los usuarios.
- Puede aplicar diferentes algoritmos de clasificación o de selección de rasgos sobre un mismo conjunto de datos.
- Define dominios en variables descriptivas, y manejo intrínseco de los dominios de datos más comunes (enteros, booleanos, reales, etc.), con posibilidad para su extensión atendiendo a las necesidades propias del problema.

CEPAR, en la actualidad, no posee algoritmos que permitan trabajar con grandes volúmenes de datos, pero admite que se le puedan agregar nuevos algoritmos para tratar este tipo de problemas, permitiendo re-utilizar los algoritmos y características que actualmente posee.

MATLAB

(abreviatura de MATrix LABoratory, «laboratorio de matrices») es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M), muy utilizado por las universidades a nivel mundial, tiene un paquete destinado al entrenamiento de distintos modelos de redes neuronales artificiales.

En la definición encontrada en Chapman (2003), MATLAB® (que abrevia MATrix LABoratory – Laboratorio de Matrices), es un programa de computadora de uso específico, que surgió como un programa para operaciones matemáticas sobre matrices y al largo de los años fue optimizado para ejecutar cálculos científicos e de ingeniería. En su libro, Chapman (2003), también destaca las ventajas y desventajas de MATLAB®.

- MATLAB® es de fácil uso y su lenguaje es ideal para el uso educacional e para el desarrollo de prototipos de nuevos programas.
- Posee plataforma independiente, con soporte para muchos sistemas computacionales, pudiendo de esta forma migrar de plataformas, conforme la necesidad del usuario.
- Existe también una amplia gama de funciones predefinidas en su biblioteca.
- En lo que dice respecto a diseños e imágenes, posee diversos comandos, y los diseños e imágenes pueden ser de cualquier dispositivo gráfico soportado por la computadora donde está MATLAB®.
- MATLAB® posee herramientas para que se construya una interface gráfica.
- La compilación de sus programas se da por la interpretación de las instrucciones en tiempo de ejecución, lo que puede demandar tiempo. Pero, el programa puede ser compilado separadamente y ser transformado en un ejecutable. E como desventajas:

- El hecho de que MATLAB® sea lento, pues el ejecuta sus programas interpretando el lenguaje y no compilando.
- E el costo elevado de su licencia.

Justificación de la selección

Después de describir las herramientas anteriormente planteadas se puede concluir que:

1. La presente investigación se encuentra enmarcada en las teorías y presupuestos del Reconocimiento Lógico Combinatorio de Patrones, por ello se consideran aceptadas las herramientas de este enfoque (MATLAB y WEKA).
2. Las herramientas que permiten añadir nuevas funcionalidades permitirán agregar los algoritmos para el cálculo de todos los Testores Típicos de escala exterior, siendo aceptadas aquellas que además estén basadas en el RLCP (WEKA).

Aun cuando resulta clara la selección de CEPAR como herramienta para implementar los algoritmos seleccionados con anterioridad, resulta oportuno realizar otras valoraciones:

- Las herramientas ACLAPIP, SELECTOR y PROGNOSIS poseen una antigüedad superior a los 10 años, y en este periodo de tiempo no han recibido mejoras para adaptarlas a nuevas tecnologías.
- Aunque WEKA permite la extensión con nuevos algoritmos, la concepción de la herramienta no cuenta con características para permitir la inserción de algoritmos y/o características del RLCP, sin que represente un cambio en su estructura interna.

Metodología, herramientas y tecnologías a utilizar

Para desarrollar la plataforma propuesta en la presente investigación, se hace necesario indagar sobre los lenguajes, herramientas y tecnologías a ser utilizados en el desarrollo de la propuesta de solución. A continuación se procede con el estudio y selección de las mismas.

Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y un soporte documental para el desarrollo de productos de software. Indicando paso a paso las actividades a realizar para lograr el producto informático deseado, indicando qué personas deben participar en el desarrollo de las actividades y qué papel deben jugar. Además, detallan la información necesaria para realizar una actividad y qué se debe producir como resultado de su ejecución. (Pressman, 2005).

Para el desarrollo de la solución propuesta se seleccionó Proceso Unificado Ágil (AUP, por sus siglas en inglés, Agile Unified Process), en su variante UCI, debido a que logra estandarizar el proceso de desarrollo de software, dando cumplimiento a las buenas prácticas que define CMMI-DEV3 v1.3.

1.8.1. Metodología de desarrollo de software AUP versión UCI

La Universidad de las Ciencias Informáticas (UCI) desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre (Yero Tarancón, 2015). A continuación, se muestra las fases de la metodología AUP-UCI:

1. Inicio: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

2. **Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además, en la transición se capacita a los usuarios finales sobre la utilización del software.
3. **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Características por escenarios

La metodología de software AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual) y existen tres formas de encapsular los requisitos los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso), surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son:

Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU. (Sánchez, 2015).

Para la presente investigación, se utilizó el escenario 1 de la metodología AUP-UCI, debido a que dicha metodología es apropiada para proyectos pequeños permitiendo disminuir las probabilidades de fracaso, por ser un producto de fácil uso utilizando cualquier herramienta. Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a 1 la cantidad de metodologías que se usaban y de más de 20

roles en total que se definían se redujeron a 11. De igual manera, también es la metodología que más utilidad tiene en la UCI por ser una versión desarrollada en dicha institución.

Lenguaje de Modelado Unificado.

Lenguaje de Modelado Unificado (UML por sus siglas en inglés) se define como un “lenguaje cuyo vocabulario y reglas se centran en la representación conceptual y física de un sistema” (UML, 2016). Se utiliza con el objetivo de visualizar, especificar, documentar y modelar los artefactos del subsistema.

Herramienta CASE.

Una herramienta CASE supone la automatización del desarrollo del software, contribuyendo a mejorar la calidad y la productividad en varios aspectos del ciclo de vida de desarrollo del software en tareas como el diseño de proyectos, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. Permite la aplicación práctica de metodologías estructuradas, las cuales al ser realizadas con una herramienta conseguimos agilizar el trabajo. Además, facilita la realización de prototipos y el desarrollo de conjunto de aplicaciones, simplifica el mantenimiento de los programas, mejora y estandariza la documentación, aumenta la portabilidad de las aplicaciones y facilita la reutilización de componentes software. También permite un desarrollo y un refinamiento visual de las aplicaciones mediante la utilización de gráficos (Somerville, 2009).

Visual Paradigm 8.0

Visual Paradigm es una herramienta CASE profesional, que utiliza el lenguaje de modelado UML. la misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software y fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (Visual Paradigm, 2015).

herramienta de prototipado

Balsamiq Mockups

Se ha empleado la herramienta **Balsamiq Mockups**[14] para realizar el documento de prototipado durante la fase de análisis, como añadido tiene la experiencia previa de uso, ya que la he usado en varias asignaturas de la carrera para la realización de varios proyectos.

Es un programa descrito en Flex y AIR creando rápidamente wireframes, posee una interfaz fácil de usar. Esta herramienta puede ser instalada en diferentes sistemas como Windows, Linux, Mac, posee funcionalidades básicas hasta la exportación a PNG o PDF (Balsamiq, 2012)

Lenguajes

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que se utiliza para escribir los programas de computadoras. Puede clasificarse teniendo en cuenta distintos criterios: lenguajes interpretados, lenguajes compilados, lenguajes de script, entre otros. El programador debe ser capaz de seleccionar el lenguaje de programación apropiado para resolver un problema bajo ciertas circunstancias específicas, de manera óptima.

Actualmente existen un conjunto de lenguajes de programación tanto para plataformas GNU/Linux, como para Windows, entre los que se destacan, Python, Java, C++, entre otros.

PHP 7.2.10

El lenguaje de programación seleccionado fue PHP en su versión 3.2.2. Ya que PHP es un lenguaje interpretado de alto nivel impregnado en páginas HTML y ejecutado en el servidor. PHP inició como una modificación a Perl a finales de 1994. Está orientado al desarrollo de aplicaciones web y permite insertar contenidos dinámicos en las páginas. El principal objetivo de PHP ha sido mejorar los mecanismos de la Programación Orientada a Objetos (POO), para solucionar las carencias de las anteriores versiones. Un paso necesario para conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes.

Características Generales del lenguaje:

- **Multiplataforma:** PHP funciona tanto en sistemas Unix o Linux con servidor web Apache como en sistemas Windows, de forma que el código generado por cualquiera de estas plataformas no debe ser modificado al pasar a la otra.
- **Ejecución en Servidor:** Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web 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.
- **Licencia de software libre:** PHP es un lenguaje basado en herramientas con licencia de software libre, lo que significa que, no hay que pagar licencias, ni se está limitado su distribución, y es posible ampliarlo con nuevas funcionalidades si así se desea.
- **Sintaxis cómoda:** PHP cuenta con una sintaxis similar a la de C, C++ o Perl. Lo más destacado ocurre a nivel semántico: el tipado es muy poco estricto. Se hace referencia específicamente a que, cuando creamos una variable no tenemos que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo.

JavaScript ES6

JavaScript es un lenguaje de programación más utilizado en Internet para añadir interactividad a las páginas Web. Es un lenguaje interpretado.(17)

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos en JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.(18)

Lenguaje marcado de hipertexto (HTML 5).

Es un nuevo lenguaje de marcado de hipertexto para presentar y estructurar el contenido en internet. Es la quinta revisión y nueva versión del estándar HTML. HTML5 ofrece nuevas características que proporcionan no sólo un rico soporte multimedia (video y audio), sino que

también mejoran el apoyo para la creación de aplicaciones web. Esta nueva tecnología puede proporcionar una solución rentable para implementar aplicaciones para apoyar diferentes dispositivos (por ejemplo, ordenador, tableta, teléfono inteligente personal) sin tener que construir aplicaciones diferentes para cada tipo de dispositivos (Gauchat, 2014).

frontend

Editor de código

Visual Studio Code 1.41.0

Visual Studio Code es un editor de código fuente ligero pero potente, siendo una aplicación de escritorio que está disponible para Windows, MacOS y Linux. Contiene soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes como C ++, C #, Java, Python, PHP, Go y tiempos de ejecución como .NET y Unity. Posee características como: terminación de código inteligente, depuración optimizada, edición rápida y potente, navegación de código y refactorización, control de fuente en el producto.(39). Se selecciona Visual Studio Code en su versión 1.18.1 como el entorno de desarrollo integrado que permitirá realizar la unificación entre el marco de trabajo de JavaScript Angular y el entorno de desarrollo Bootstrap que se utilizan para la codificación del software en el lado del cliente.

Entorno de desarrollo Integrado (IDE)

Un Entorno de Desarrollo Integrado es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, unas herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones. (Rodríguez., 2010).

backend

PHPStorm v 2019.3.1

Es un potente IDE especialmente diseñado a fin de proporcionar a los desarrolladores de HTML, JavaScript y PHP todas las herramientas necesarias para su trabajo. PhpStorm proporciona un editor de código enriquecido e inteligente para PHP con resaltado de sintaxis, configuración extendida de formateo del código, comprobación de errores sobre la marcha y finalización de código inteligente.

A continuación, algunas características de PhpStorm:

- Establecer puntos de interrupción condicionales con paso inteligente, lo que te permite seleccionar un método específico para ejecutar desde la cadena de llamadas.
- Inspeccionar variables locales pertinentes para el contexto y los relojes definidos por ti, incluyendo los arreglos de discos y los objetos complejos y editar los valores sobre la marcha.
- Evaluar una expresión en el tiempo de ejecución.
- Depurar una página en varias sesiones simultáneamente.
- Mantener una sesión de depuración mientras te desplazas entre páginas.

Marco de Trabajo.

Según el autor Rodríguez O. ,2010, un marco de trabajo es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática, proveyendo una metodología de trabajo que sistematiza y facilita la generación de formularios, funciones y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación.

En el desarrollo de software un entorno o marco de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software Típicamente, puede incluir soporte de programas bibliotecas y un lenguaje interpretado entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Symfony

Framework

Symfony

Symfony v2.8.4 es un framework de tipo full-stack 3construido con varios componentes independientes y librerías que se utilizan para crear aplicaciones y sitios web en PHP basado en el patrón Modelo Vista Controlador. Su código, todos los componentes y librerías que incluye, se publican bajo la licencia MIT 4de software libre. También la documentación del proyecto es libre e incluye varios libros y decenas de tutoriales específicos.

Symfony permite establecer los parámetros de configuración de las aplicaciones a través de variables de entorno del propio servidor para evitar el uso de contraseñas en archivos de configuración. Dispone de un plan de lanzamientos predecible, con versiones estables mantenidas durante tres años. Symfony incluye varias herramientas gráficas y de consola para depurar fácilmente los errores que se produzcan en las aplicaciones (symfony, 2016).

Symfony es utilizado en la presente investigación para el desarrollo del subsistema, ya que este framework de desarrollo es el que utiliza el proyecto y actualmente es estable. Además, con la utilización del patrón Modelo Vista Controlador permite a los desarrolladores y los diseñadores trabajar simultáneamente, incluyendo la capacidad de hacer prototipos rápidos, hacer cambios en una parte del subsistema sin afectar a los demás. Permite que el subsistema sea compatible con bibliotecas e infraestructuras que existen y que se pueda adaptar a entornos de negocio en cambio permanente. También es compatible con la mayoría de los sistemas gestores de base de datos en el caso de la presente investigación PostgreSQL y permite la utilización de lenguajes como HTML5, PHP7 y CSS3 (LibrosWeb, 2016).

React.js

React es un marco de trabajo de JavaScript, creado para resolver los retos involucrados al desarrollar interfaces complejas con un conjunto de datos que cambian con el tiempo. Es un concepto diferente cuando se trata del desarrollo web en general. Es un cambio de los flujos de trabajo generalmente aceptados y las mejores prácticas, ya que afronta las convenciones que se han convertido en los estándares para las buenas prácticas del marco de trabajo de JavaScript, al

introducir muchos nuevos paradigmas y al cambiar el status de lo que se necesita para crear aplicaciones de JavaScript e interfaces de usuarios escalables y mantenibles. React no presenta como patrón de diseño Modelo Vista Controlador (MVC) como otros muchos marcos de trabajo, ya que es una representación más simple, siendo solo el componente Vista de este patrón de diseño. Está hecho con componentes declarativos que describen una interfaz. No utiliza ningún enlace de datos observable al crear una aplicación. Asimismo, es fácil de manipular, ya que se pueden tomar los componentes y combinarlos para crear componentes personalizados que funcionen como el desarrollador desee.(22)

Para evitar la complejidad de las interacciones y el procesamiento posterior requerido, React realiza un renderizado completo de la aplicación. Es también declarativo. Mantiene un flujo de trabajo simple. Resuelve esto dando al desarrollador un DOM virtual para representar en lugar del DOM real. Encuentra la diferencia entre el DOM real y el DOM virtual y lleva a cabo el número mínimo de operaciones DOM requeridas para alcanzar el nuevo estado. (23)

Sistema Gestor de Base de Datos

Definimos un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (Data Base Managment System) como una colección de datos relacionas entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos.

Se trata de un conjunto de programas no visibles al usuario final que se encargan de la privacidad, la integridad, la seguridad de los datos y la interacción con el sistema operativo. Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. Cualquier operación que el usuario hace contra la Base de Datos está controlada por el gestor.

PostgreSQL 10.0

PostgreSQL es un sistema de administración de bases de datos de propósito general y objeto-relacional, el sistema de base de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un

amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, tcl y python).. Es un software gratuito y de código abierto. Su código fuente está disponible bajo la licencia PostgreSQL, una licencia libre de código abierto.(37)

PostgreSQL es el primer sistema de administración de base de datos que implementa la característica de control de concurrencia de múltiples versiones (MVCC), incluso antes de Oracle. Es un sistema de administración de bases de datos relacionales de objetos de propósito general. Permite agregar personalizado a funciones desarrolladas usando diferentes lenguajes de programación como Java. Está diseñado para ser extensible. Se pueden definir personalizaciones en los tipos de datos, tipos de índice, idiomas funcionales.(37). Se selecciona PostgreSQL en su versión 9.4, a partir de las ventajas que brindan descritas anteriormente, como el gestor de base de datos que se utilizará para el manejo y obtención de los datos referentes al sistema a desarrollar.

MySQL v 3.4

MySQL es un gestor de base de datos que tiene la capacidad de almacenar gran cantidad de información y numerosos tipos de datos, es de código abierto, multi-hilos, lo que le permite aprovecharse de los sistemas multiprocesadores, es multiplataforma y multiusuario, además administra la base de datos de forma relacional.

Cuenta a su disposición con una buena cantidad de APIs en numerosos lenguajes, que juega además con su gran portabilidad entre sistemas. MySQL llega a soportar hasta 32 índices por tabla en las cuales mantiene un alto nivel de seguridad, evidenciado en la gestión de usuarios y contraseñas que provee.

Inicialmente se pensó en utilizar este gestor, pero por cuestiones que se expondrán en el siguiente epígrafe y una restricción de uso de software, no se hará uso de este gestor de base de datos, además, porque solo las viejas versiones de MySQL son gratis. (MySQL.com, 2012)

Conclusiones parciales

- ✓ La investigación acerca de las tecnologías homólogas brindó un mayor entendimiento y conocimiento al desarrollador sobre las tendencias de diseño web, lo cual permitió la adecuada selección de las tecnologías y herramientas a utilizar en el desarrollo del sistema.
- ✓ La correcta selección de las herramientas y tecnologías de desarrollo garantiza un adecuado desarrollo del software.
- ✓ La selección de la metodología de desarrollo de software favorece el adecuado diseño del sistema, así como la realización de cada una de las fases del ciclo de vida del software. Mediante la cual, se documentará todo el ciclo de desarrollo de la investigación.

CAPÍTULO 2. Descripción de la solución Propuesta

En este capítulo se describen las características y los procesos que contienen toda la información relacionada con el sistema propuesto. Se recogen aspectos importantes de todo el entorno de trabajo para un mayor entendimiento del problema planteado. Se detallarán los distintos pasos de la metodología propuesta en el capítulo anterior (AUP-UCI) para el desarrollo de la aplicación. Esto

incluirá la elaboración del modelo del dominio, la exposición de los requerimientos de la plataforma; definición de Actores y relaciones entre ellos, los Diagramas de Caso de Uso y las descripciones textuales de los mismos teniendo en cuenta la acción del actor y la respuesta del sistema; el patrón arquitectónico y finalmente los Diagramas de Clases del Diseño y del Modelo de datos.

En general el capítulo aborda todos los temas relacionados con la ubicación del problema y necesidades que se presentan para resolverlo sin tener definida una solución.

Análisis y diseño del sistema

Modelo de Dominio

Un Modelo del Dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Muchos de los objetos del dominio o clases pueden obtenerse de una especificación de requisitos. [18]

El modelo de dominio tiene como objetivo fundamental visualizar a los usuarios los principales conceptos que se manejan en el dominio a desarrollar a través de la comprensión y descripción de las clases más importante del sistema.

Conceptos del Dominio

Para una mejor comprensión del Diagrama del Modelo de Dominio estructurado en el siguiente epígrafe, a continuación se proporciona un marco conceptual con las definiciones identificadas. Estas son:

Herramienta Educativa: Entorno de aprendizaje virtual donde el estudiante puede visualizar los contenidos agregados por el profesor y la resolución automática de la clasificación según la base de datos cargada por él mismo.

Usuario: Existen tres tipos de usuarios definidos por la Herramienta que podrán hacer uso de los servicios de la misma, estos son: Profesor, Administrador y Estudiante. Estos pueden consumir los

servicios de la herramienta, ya que cualquiera puede hacer uso de los archivos almacenados en el Sistema para el estudio del tema de la asignatura.

Profesor: Profesional académico que domina el tema de la asignatura.

Administrador: Profesional académico que domina el tema de la asignatura y realiza el diseño instruccional de los contenidos. Los Administradores también son profesores. Solo los Administradores tendrán la posibilidad de gestionar los recursos y los usuarios del sistema.

Plantilla: es una estructura que crea el autor a través de la cual se organizan los contenidos de la experiencia de aprendizaje.

Recurso Educativo: Cualquier entidad digital que pueda ser usada en el aprendizaje. Representación digital de texto, imágenes, audio, video, etc.

Metadatos: Descripciones de los recursos.

Estudiante: Persona que interacciona con el sistema o aplicación con el fin de conseguir unos objetivos determinados de aprendizajes.

Archivo BD: un archivo de base de datos en el formato CSV que será considerado como matriz de aprendizaje(MA).

Diagrama del Modelo de Dominio

El siguiente diagrama muestra la relación entre los conceptos identificados anteriormente descritos en el problema.

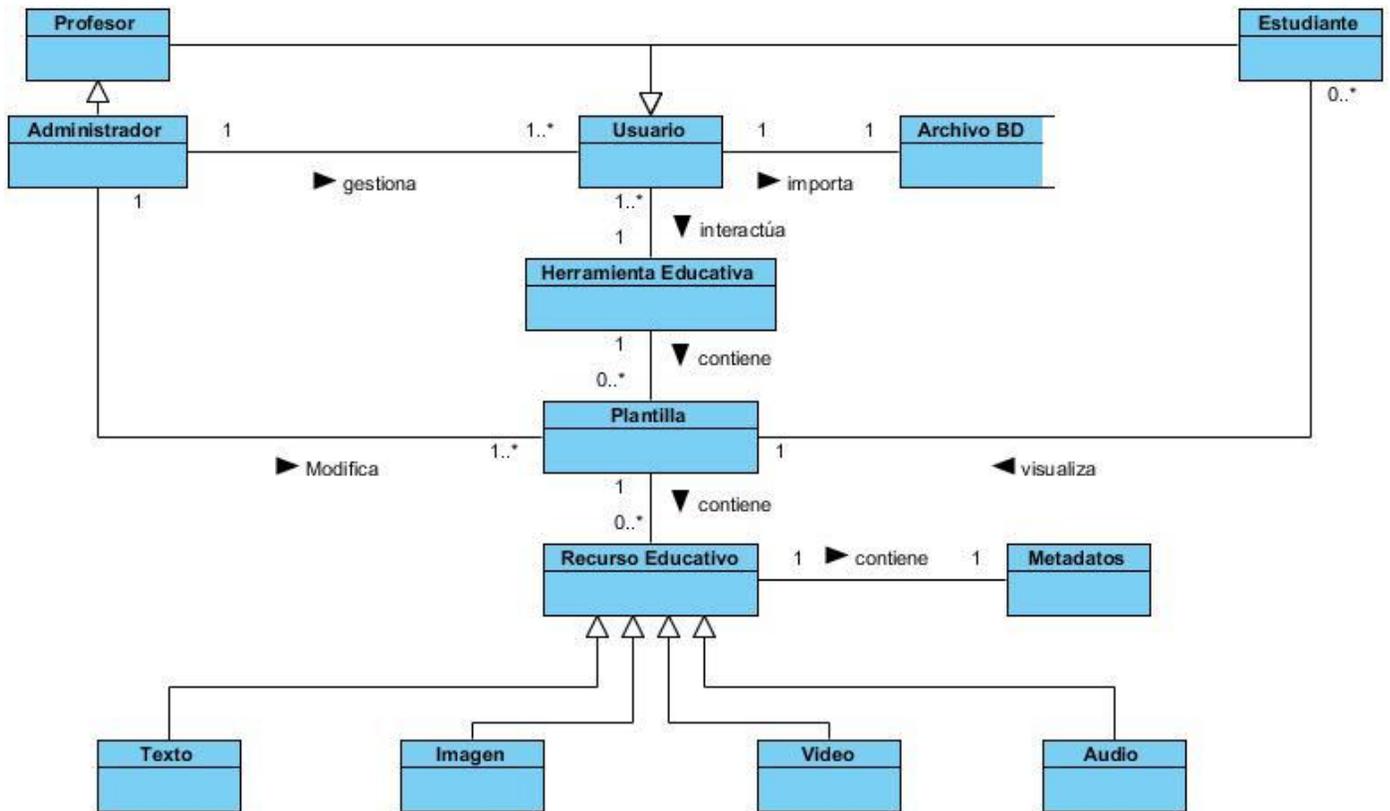


Figura 3 Diagrama del Modelo de Dominio

Requisitos del Sistema

Requisitos Funcionales

Los requerimientos funcionales serán las capacidades funcionales que la aplicación debe cumplir, es decir, una serie de actividades que serán objeto de automatización más adelante.

Sesión

Referencia	Función
R1	Autenticar usuario
R2	Cerrar sesión
R3	Modificar plantilla educativa

Gestionar Usuario

Referencia	Función
R4	Mostrar usuarios
R5	Eliminar usuario
R6	Buscar usuario

Gestionar Recurso Educativo

Referencia	Función
R7	Cargar Recurso Educativo
R8	Visualizar Recursos Educativos
R9	Modificar Recurso Educativo
R10	Eliminar Recurso Educativo
R11	Seleccionar Recurso Educativo

Principal

Referencia	Función
R12	Importar Base de Dato (Formato CSV)
R13	Seleccionar tipo de rasgo
R14	Seleccionar función de comparación
R15	Calcular MD
R16	Calcular MB
R17	Identificar rasgos
R18	Descargar Recurso

Requisitos Generales

Referencia	Función
R19	Permitir retornar a la parte superior de la página clicando en un botón.
R20	Mostrar información de la herramienta

Requisitos no Funcionales

Los requerimientos no funcionales identifican un conjunto de propiedades y cualidades que el producto debe tener, es decir, con los requisitos no funcionales se mencionan las características principales que harán al producto atractivo, usable, rápido y confiable.

Apariencia o interfaz externa.

RNF 1. Se debe garantizar que la interfaz de la aplicación sea lo más atractiva y clara posible para el desarrollador.

RNF 2. Se debe garantizar que la interfaz sea amigable y de fácil comprensión para los usuarios que por primera vez la utilicen.

RNF 3. La aplicación debe proveer al usuario una gran versatilidad en la presentación de la información en las pantallas, en cuanto a disponibilidad de los recursos y organización de los mismos.

RNF 4. Diseño orientado a crear una visión agradable para el usuario y con una navegación sencilla.

RNF 5. Diseño intuitivo que permita a los usuarios usar cómoda y fácilmente la aplicación.

RNF 6. El sistema debe tener una interfaz con colores adecuados al contenido, predominando un estilo de diseño sencillo.

RNF 7. Se debe informar al usuario sobre el estado de cualquier acción que ejecute, informándole mediante notificaciones el resultado de estas, tanto si es error, alerta o éxito.

Hardware.

Teniendo como base los requisitos mínimos de hardware de las herramientas y tecnologías seleccionadas para obtener un buen rendimiento se plantea:

RNF 8. Para uso del cliente: Procesador Intel Celeron a 1.8 GHZ (Gigahercio) o superior, 256 MB (Megabyte) de RAM ((Random Access Memory o Memoria de Acceso Aleatorio) o superior y 650 MB de espacio libre en disco.

RNF 9. Para uso del servidor: Procesador Intel Dual CPU 1.86 GHz o superior, 4 GB de RAM o superior y 40 GB de espacio libre en disco.

Software.

RNF 10. El sistema se implementará con tecnología PHP 7.2.10.

RNF 11. Se requiere PostgreSQL 11.0 como servidor de bases de datos.

RNF 12. Servidor Web Apache.

Seguridad.

RNF 13. Necesidad de autenticación para acceder a las funcionalidades de la aplicación.

La seguridad del sistema desarrollado está basada en niveles de acceso sobre las funcionalidades y la información manejada por el sistema. El principio que determina la seguridad en el sistema es el siguiente:

RNF 14. La seguridad se establecerá por roles que serán asignados a los usuarios que interactúen con el sistema, garantizando de esta forma que la información almacenada solo sea modificada y/o visualizada por los usuarios autorizados.

Usabilidad.

RNF 15. Debe ser de fácil manejo o manipulación aún para personas que no tengan muchos conocimientos de trabajo en computadoras y ambientes WEB.

RNF 16. El idioma definido para mostrar los mensajes y textos de la interfaz es el español.

RNF 17. La navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario empleándose la regla de los 3 clicks (3 clicks rules) la cual plantea que no se debería llegar a ninguna información clave de un sitio web en más de tres clicks (39).

RNF 18. El sistema debe ser escalable para que al agregar nuevas funcionalidades no sean afectadas las que ya están funcionando.

RNF 19. El sistema debe poseer una interfaz fácil de utilizar para cualquier tipo de usuario con conocimientos básicos de informática y en el manejo de ordenadores.

Disponibilidad.

RNF 20. El sistema debe estar disponible las 24 horas del día en el período que se impartirán las clases, este período será definido por el administrador.

Portabilidad.

RNF 21. La aplicación debe funcionar en varias plataformas, siendo posible su acceso a través de cualquier navegador Web.

Rendimiento.

RNF 22. Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de

elaboración en la Web, que faciliten el rápido acceso a sus páginas. La herramienta propuesta debe ser rápida y el tiempo de respuesta no debe ser mayor de 2 segundos, adecuado a la rapidez con que el usuario requiere la respuesta a su acción.

Navegación.

RNF 23. Desde cualquier pantalla de la aplicación se podrá cerrar la sesión. Para ello el Sistema mostrará un aviso al usuario de la acción que estará a punto de realizar.

RNF 24. Se requiere que desde cualquier pantalla en que se encuentre el usuario, se pueda acceder a los otros módulos contenidos en la aplicación.

Ayuda:

RNF 25. Proporcionar una ayuda para orientar al usuario sobre el funcionamiento de la herramienta.

Modelo de Casos de Uso del Sistema

En esta sección se muestran los posibles actores del sistema que van a interactuar con los casos de usos definidos a partir de los requisitos funcionales antes expuestos. Lo cual posibilitará un mejor entendimiento de las funcionalidades que se requieren del sistema y darán un resultado de valor para los actores identificados.

Descripción de los actores del sistema

Se le llama actor a toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Los actores pueden representar roles jugados por usuarios humanos, hardware externo, u otros sujetos. Un actor no necesariamente representa una entidad física específica, sino simplemente una faceta particular (es decir, un "rol") de alguna actividad que es

relevante a la especificación de sus casos de uso asociados. Así, una única instancia física puede jugar el rol de muchos actores diferentes y, asimismo, un actor dado puede ser interpretado por múltiples instancias diferentes. (Kruchten, 2003)

A continuación, se definen y describen los actores que intervienen en el proceso.

Tabla 2. Descripción de los Actores

Actor	Descripción
Usuario	Es la persona que accede al sistema mediante el proceso de autenticación. Existen tres tipos de usuarios definidos por la Herramienta que podrán hacer uso de los servicios de la misma, estos son: Profesor, Administrador y Estudiante.
Profesor	Profesional académico que domina el tema de la asignatura.
Administrador	Es quien posee la mayoría de los permisos. Es quien gestiona los recursos educativos y los usuarios del sistema. El Administrador también es un profesor.
Estudiante	Es la persona que tiene acceso a las funcionalidades públicas y educativas, este puede interactuar con los recursos y contenidos existentes en la plantilla del sistema.

Diagrama de Casos de Uso del Sistema

Un diagrama de casos de uso del sistema explica gráficamente un conjunto de casos de uso (CU) del sistema, los actores y cómo se relacionan entre ellos. El diagrama tiene por objetivo ofrecer una clase de diagrama contextual que permite conocer rápidamente los actores externos de un sistema y las formas básicas en que lo utilizan (Larman, 2003).

Tabla 3. Agrupación de requisitos por caso de uso

Nombre del caso de uso	Referencia a requisitos
Autenticar usuario	RF: R1
Cerrar sesión	RF: R2
Modificar plantilla	RF: R3

Gestionar usuario	RF: R4,R5,R5
Gestionar recurso educativo	RF: R7,R8,R9,R10
Seleccionar Recurso Educativo	RF: R11
Importar Base de Dato	RF: R12
Seleccionar tipo de rasgo	RF: R13
Seleccionar función de comparación	RF: R14
Calcular MD	RF: R15
Calcular MB	RF: R16
Identificar rasgos	RF: R17
Descargar Recurso	RF: R18
Retornar a la parte superior de la página	RF: R19
Mostrar información de la herramienta	RF: R20

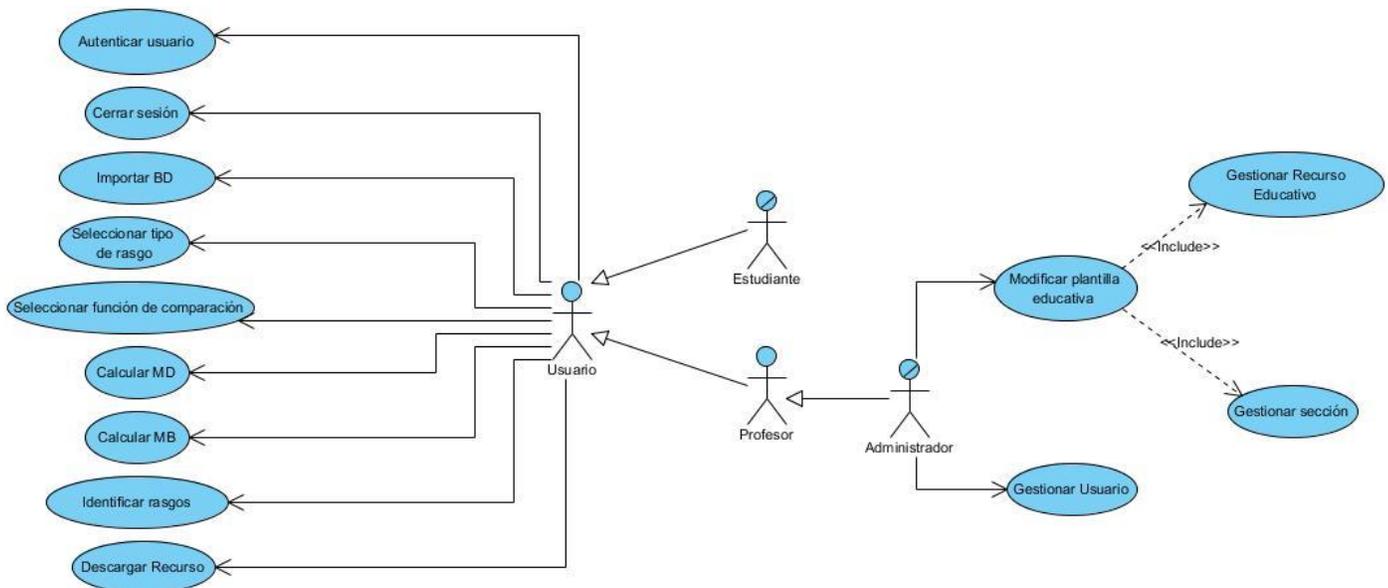


Figura 3 Diagrama de Casos de Uso del Sistema

Descripción de los Casos de Uso del Sistema

La descripción textual de los casos de uso permitirá una comprensión detallada del funcionamiento del sistema.

CU 3. Modificar plantilla educativa

Tabla 4. Descripción del caso de uso: Gestionar recurso educativo

Objetivo	Modificar/actualizar la plantilla
Actores	Administrador (Inicia): Modificar plantilla educativa
Resumen	El caso de uso se inicia cuando el Administrador escoge la opción “Modificar plantilla educativa” . Dentro del mismo puede Gestionar un recurso (subir, modificar o eliminar un recurso), como también personalizar la plantilla de los recursos mediante Gestión de secciones (Adicionar, eliminar, modificar y cambiar nombre de secciones). Este finaliza cuando se han efectuado las acciones en la plantilla.
Complejidad	<i>Alta</i>
Prioridad	<i>Auxiliar</i>
Precondiciones	El usuario con rol Administrador deberá estar autenticado en el sistema y debe existir conexión a la Base de Datos.
Postcondiciones	<i>Se gestionan los recursos educativos.</i>
Flujo de eventos	
Flujo básico “Gestionar recurso educativo”	

	Acción del Actor	Respuesta del Sistema
1.	Selecciona de la interfaz principal la opción “Modificar plantilla educativa” .	
2.		<p>2.1- Muestra una interfaz con un formulario listo para la inserción de los datos del tema permitiendo realizar varias acciones con los recursos educativos:</p> <ul style="list-style-type: none"> • Cargar recurso educativo (Ver Flujo básico de eventos). • Modificar recurso educativo (Ir a la Sección 2 Modificar recurso educativo). • Eliminar recurso educativo (Ir a la Sección 1 Eliminar recurso educativo). <p>2.2- Siendo los siguientes datos requeridos en el formulario:</p> <ul style="list-style-type: none"> • titulo (campo de texto) (campo obligatorio) • contenido (por un editor de texto) (campo obligatorio). Es el campo donde se insertan los recursos educativos.
3.	El Autor introduce los datos requeridos.	
4.	El Autor decide cargar un recurso educativo en el campo del contenido accediendo mediante un clic izquierdo en la zona del texto donde necesita hacer la inserción.	

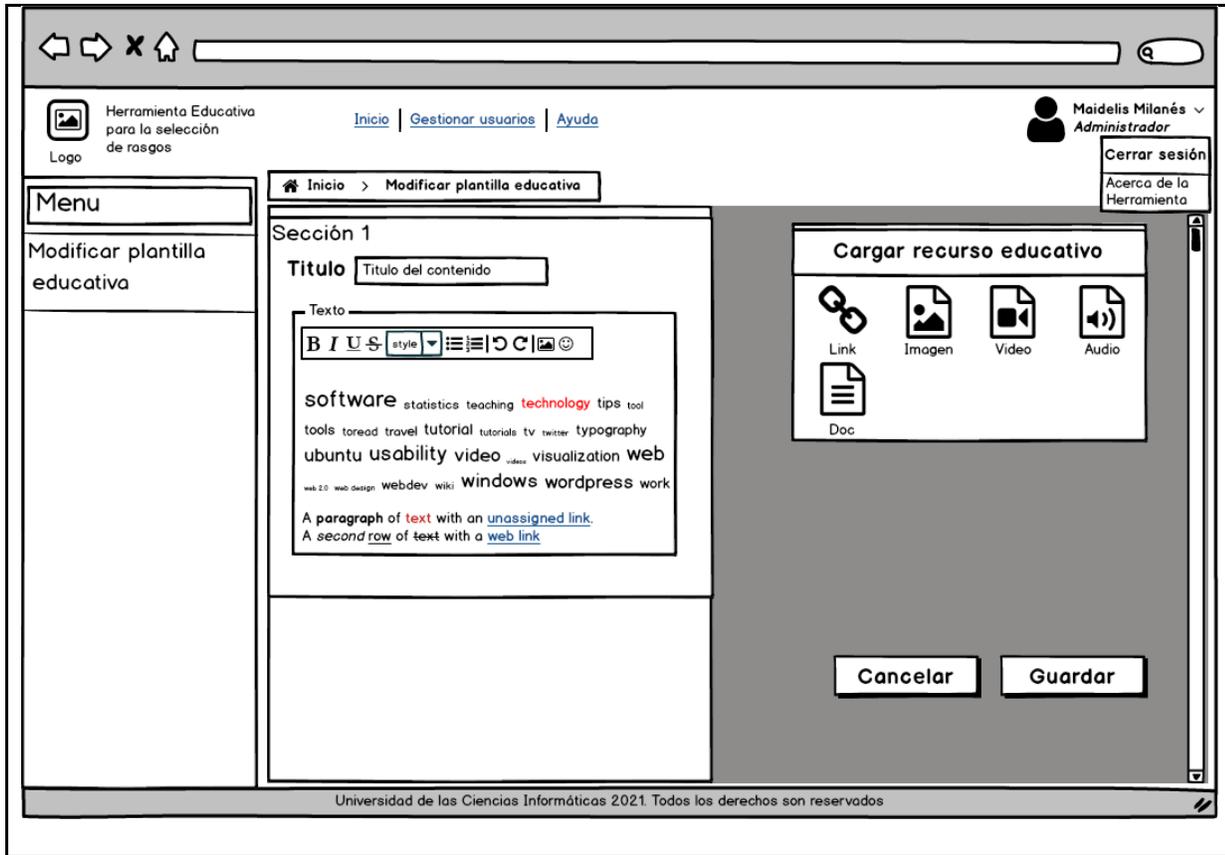
5.	En seguida hace otro clic izquierdo sobre el ícono del tipo de recurso que necesite cargar.	
6.		5.1- La herramienta abre una nueva interfaz con el acceso a los directorios del ordenador para poder seleccionar el recurso que se desea junto con un campo de texto para introducir la descripción del recurso.
7.	El Autor Selecciona el recurso que desea, escribe su descripción y selecciona la opción Insertar.	
8.		7.1- La herramienta carga el recurso en la zona seleccionada del campo del contenido, y brindando del mismo tres opciones que se mostraran a través de un clic derecho sobre el recurso a lo cual se desee o modificarlo, o eliminarlo o cambiar su descripción. a) Modificar recurso, ver Sección 1 . b) Eliminar recurso, ver Sección 2 . a) Cambiar descripción del recurso, ver Sección 3
Flujos alternos		
Flujo básico		
9.	Acción del Actor	Respuesta del Sistema
10.		
11.		
Sección 1 “Modificar recurso”		

1.	El Autor decide modificar el recurso accediendo mediante un clic derecho, y selecciona la opción “Modificar recurso” .	
2.		La herramienta abre una nueva interfaz con el acceso a los directorios del ordenador para poder seleccionar el recurso que se desea junto con un campo de texto para introducir la descripción del recurso.
3.	El Autor Selecciona el nuevo recurso que desea, modifica su descripción y selecciona la opción Insertar.	
4.		La herramienta reemplaza el recurso viejo por este nuevo.
Flujos alternos		
Acción del Actor		
Respuesta del Sistema		
5.		
6.		
7.		
Sección 2 “Eliminar recurso”		
Flujo básico		
8.		
9.	El Autor decide eliminar el recurso accediendo mediante un clic derecho, y selecciona la opción “Eliminar recurso” .	
10.		Elimina el recurso.

11.	El autor decide agregar una nueva sección mediante un clic derecho sobre la sesión existente.	
12.		<p>9.1- La herramienta verifica que los datos de la sección existente se hayan completado.</p> <p>9.2- Visualiza el área donde definir la estructura de la plantilla, brindando en la misma tres opciones, que se mostraran a través de un clic derecho sobre la sesión a la cual se desee o agregarle una nueva, o eliminarlo o cambiar su nombre.</p> <p>a) Agregar sección, ver Sección 1.</p> <p>b) Eliminar sección, ver Sección 2.</p> <p>a) Cambiar el nombre de la sección, ver Sección 3</p>
Sección 1 “Agregar sección”		
13.	El Autor decide agregar una sesión accediendo mediante un clic derecho, y selecciona la opción “Agregar sección” .	
14.		<p>11.1- La herramienta adiciona la nueva sección.</p> <p>11.2- La herramienta actualiza el manifiesto (XML) donde se almacena la estructura de la plantilla.</p>
Sección 2 “Eliminar sección”		

15.	El Autor decide eliminar una sección, para lo cual accede mediante un clic derecho, y selecciona la opción “Eliminar sección”	
16.		13.1- La herramienta elimina la sección. 13.2- La herramienta actualiza el manifiesto (XML) donde se almacena la estructura de la plantilla.
Sección 3 “Cambiar el nombre de la sección”		
17.	El Autor decide cambiar el nombre a una sección, para lo cual, lo selecciona a través de un clic derecho, y escoge la opción “Cambiar nombre” .	
18.		15.1- La herramienta habilita el campo nombre del nivel para que el Administrador pueda modificarlo.
19.	El Autor introduce el nombre que desee.	
20.		18.1- La herramienta cambia el nombre de la sección de organización. 18.2- La herramienta actualiza el manifiesto (XML) donde se almacena la estructura de la plantilla.
21.	El Autor da clic izquierdo en “Guardar” .	

22.		19.1 Valida los datos introducidos.
23.		20.1- La herramienta guarda los cambios hechos en la plantilla, aplica estos cambios en la pantalla de inicio y muestra un mensaje informando al usuario que se han guardado exitosamente los cambios. Termina el CU.
Flujos alternos		
11a. No se agrega la sesión.		
	Acción del Actor	Respuesta del Sistema
1.	El Autor no agrega ninguna sesión	
2.		2.1- La herramienta conserva la estructura inicial de la plantilla conformada por una organización y un ítem.
Sección 1: "Nombre"		
Flujo básico <Nombre del flujo básico>		



CU 1. Autenticarse

Objetivo	<i>Acceder al sistema</i>
Actores	<i>Usuario</i>
Resumen	El caso de uso se inicia cuando el usuario escoge la opción Descargar Recurso ofreciéndole la posibilidad de almacenar el mismo en su máquina y finaliza luego de haber descargado el Recurso Educativo.
Complejidad	<i>[Alta, Media, Baja]</i>
Prioridad	<i>[Crítico, Secundario, Auxiliar, Opcional]</i>

Precondiciones	<i>[La precondición declara de qué debe estar seguro el sistema antes de autorizar el inicio del caso de uso. Esto no debe ser chequeado de nuevo mientras dura el caso de uso. Usualmente tener una precondición indica que otro caso de uso está o se ha ejecutado. Una precondición debe escribirse como una afirmación simple en la forma verbal antepresente. Por ejemplo: El cliente <u>ha sido</u> validado. (COCKBURN '00).]</i>	
Postcondiciones	<i>[Establece qué intereses deben ser satisfechos cuando se haya terminado exitosamente el caso de uso. Debe escribirse como una afirmación simple en pasado, que debe ser cierta al terminar el caso de uso. (COCKBURN 2000).]</i>	
Flujo de eventos		
Flujo básico <Autenticarse>		
	Acción del Actor	Respuesta del Sistema
24.	<i>[El caso de uso debe iniciarse siempre por el actor. Debe omitirse comenzar la descripción de los eventos con los términos “El actor hace tal cosa”, o “El sistema hace tal cosa”, pues las columnas ya identifican quién realiza cada acción. Debe comenzarse con un verbo de acción. Ejemplo: Valida que los datos sean correctos.]</i>	
25.		<i>[Para las secciones debe de colocarse una lista de acciones que el sistema permite, seguido del nombre de la sección. Inmediatamente después el flujo básico debe describir el flujo de eventos de la primera. Por ejemplo: Permite realizar varias acciones con una persona: - Adicionar una nueva persona. - Eliminar una persona. Ver Sección 1:”Eliminar persona”.]</i>
26.		<i>[El último evento del caso de uso debe ser: Termina el caso de uso.]</i>
Flujos alternos		
Nº Evento <Condición que dio lugar a la extensión> <i>[Ejemplo: 1a.Los datos son incorrectos.]</i>		

	Actor	Sistema
3.		
4.		
Sección 1: "Nombre"		
Flujo básico <Nombre del flujo básico>		
	Actor	Sistema
Flujos alternos		
Nº Evento <Condición que dio lugar a la extensión>		
	Actor	Sistema
1.		
2.		
Relaciones	CU Incluidos	<p><i>[Listado de los casos de uso y/o salidas del sistema incluido por el caso de uso. Deben de referenciarse en el cuerpo de la especificación como:</i></p> <p><i><<Evento que se ejecuta>>. <u>Ver CU <<Nombre del caso de uso>></u></i></p> <p><i>Ejemplo: Insertar usuario. <u>Ver CU Insertar Usuario.</u></i></p>
	CU Extendidos	<p><i>[Listado de los puntos de extensión (PE) de los casos de uso que se extienden o que se publican por el caso de uso.</i></p> <p><i>Para quien publique el PE:</i></p> <p><i><<Nombre del PE>>: Paso <<Nº>> del Flujo Básico.</i></p> <p><i>Ejemplo: Registrar usuario: Paso 5 del Flujo Básico.</i></p> <p><i>Para quien extienda el PE:</i></p>

		<<Nombre del PE>> en el <<Nombre del CU>>. <i>Ejemplo: Registrar usuario en el CU Administrar permisos.]</i>
Requisitos no funcionales		<i>[Son los requisitos no funcionales específicos para el caso de uso]</i>
Asuntos pendientes		<i>[Posibles mejoras al caso de uso.]</i>

Patrón arquitectónico

Los patrones arquitectónicos son patrones de diseño de software que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor. (51) Para el desarrollo del componente se utiliza el patrón arquitectónico Modelo-Vista-Controlador (MVC) en el cual está basado Symfony, aunque de la versión 2 del framework su creador dijera que Symfony 2 no es MVC, pues sólo proporciona herramientas para la parte del controlador y de la vista, la parte del modelo es responsabilidad del ORM Doctrine.

Resulta esencial conocer qué es una arquitectura MVC y cómo se aplican sus principios fundamentales a las aplicaciones en Symfony 2. El patrón MVC define tres niveles: **El modelo:** representa la información relacionada con el dominio de la aplicación, es decir, su lógica del negocio. Abstrae la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes del tipo de gestor de base de datos utilizado. En la solución se utiliza Doctrine, una biblioteca cuyo objetivo es brindar poderosas herramientas para el acceso a una base de datos relacional. El ORM de Doctrine permite asociar objetos a una base de datos tal como MySQL,

PostgreSQL o Microsoft SQL. El componente mediante las funcionalidades de Doctrine, puede recuperar objetos completos de la base de datos, tales como el DropboxFile.

La vista: transforma el modelo en interfaces de usuario permitiendo la interacción con el sistema. Solo se encarga de mostrar información. Twig es un motor y lenguaje de plantillas para PHP muy rápido y eficiente. Symfony 2 recomienda utilizar Twig para crear las plantillas de la aplicación. La sintaxis de Twig se ha diseñado para que las plantillas sean concisas y muy fáciles de leer y de escribir. En el componente se muestran varias plantillas, un ejemplo de estas es index.html.twig, que se utiliza para listar los archivos almacenados en la nube.

El controlador: se encarga de procesar las peticiones realizadas desde el navegador realizando los cambios necesarios en el modelo o en la vista. Este se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comando, etc.). En el componente los controladores son funciones PHP que toman información de la petición HTTP y construyen una respuesta HTTP como un objeto de Symfony 2. Los controladores tienen la lógica que el componente necesita para reproducir el contenido de las páginas. La clase controladora DropboxController.php tiene varias acciones, por ejemplo, addNewResource que se encarga de descargar un archivo, adicionar la información en la entidad correspondiente y mostrar un mensaje de confirmación.

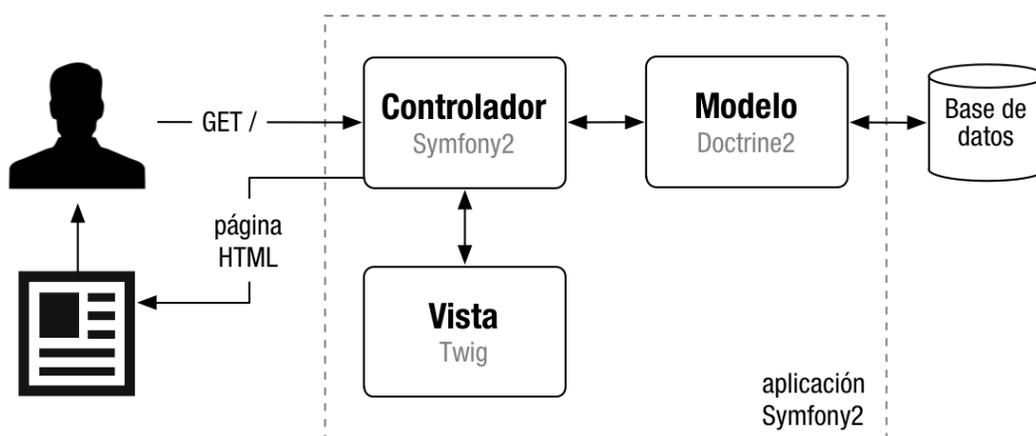


Figura 4 Esquema simplificado de la arquitectura MVC de Symfony2.

2.7 Patrones de diseño empleados

Symfony sigue la mayoría de las mejores prácticas y patrones de diseño para la web (44). Para dar solución a la implementación de los requisitos funcionales del componente planteados en el anterior capítulo se aplican algunos patrones que guían el proceso de desarrollo. Estos constituyen la solución de un problema determinado y se pueden aplicar en diferentes contextos, su finalidad no es expresar nuevas ideas en el diseño sino resolver los problemas mediante una solución ya probada y fiable (52).

2.7.1 Patrones GRASP Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades, según siglas traducidas al español), describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones, y definen un grupo de principios fundamentales para diseñar eficazmente un software. Algunos de los utilizados en la solución que son implementados por Symfony se describen a continuación:

- **Experto:** Expresa que los objetos realizan funciones de acuerdo a la información con que cuentan. Esto trae como beneficio que se conserve el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Además, favorece el bajo acoplamiento. El comportamiento se distribuye entre las clases que cuentan con la información requerida alentando con ello definiciones de clases sencillas que son fáciles de comprender y mantener. Se pone de manifiesto cuando la clase controladora `DropboxController.php` es llamada a responder determinado pedido y esta cuenta con la información necesaria para satisfacer la petición.
- **Creador:** Permite asignar quién debería ser el responsable de la creación de una nueva instancia de alguna clase. Guía la asignación de responsabilidades relacionadas con la creación de objetos. El propósito general de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Se manifiesta en gran parte de la solución al ser creadas instancias de las clases necesarias para ejecutar las funcionalidades, ejemplo de esto cuando se crea un nuevo recurso, se hace necesario instanciar la clase `resource.php` que se encuentra en

LearningResourceBundle.

- **Alta Cohesión:** El objetivo de este patrón es asignar responsabilidades de tal forma que la cohesión siga siendo alta. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Es utilizado en la inyección de dependencias de la clase controladora, cuando se pasa (inyecta) a esta clase todos los objetos que necesita (dependencias) ya creados y configurados.

- **Controlador:** El propósito de este patrón es asignar la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. El controlador maneja todas las peticiones web en el componente. Se evidencia en la solución cuando la clase controladora convierte una petición generada por el usuario en la respuesta de un método que utiliza una clase específica del SDK de Dropbox, convirtiéndose así en un controlador del flujo de eventos generados en el componente.

- **Bajo Acoplamiento:** El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios. Este patrón especifica cómo dar soporte a una dependencia escasa y a un aumento de la reutilización. Se pone de evidencia al utilizar la inyección de dependencias, que además de crear una alta cohesión, permite crear un bajo acoplamiento entre las clases, pues se asignan responsabilidades a clases específicas en el sistema y se utilizan según sea necesario.

2.7.2 Patrones GOF

Los patrones GOF se descubren como una forma indispensable de enfrentarse a la programación a raíz del libro “**Design Patterns, Elements of Reusable Software**”, a partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro (GOF, **gang of four**), haciendo referencia a los cuatro autores de este libro. (53) Algunos utilizados por el **framework** de desarrollo Symfony presentes en la solución son:

- **Decorador (Envoltorio):** Añade funcionalidad a una clase, dinámicamente. El archivo layout.php, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página.
- **Fachada:** Symfony, mediante el sistema de configuración de archivos .yml, implementa este patrón permitiendo acceder a diferentes configuraciones del **framework** desde un único lugar. Además de que el acceso a la aplicación es a través del controlador frontal que implementa este patrón.
- **Adapter (Adaptador):** Symfony da la posibilidad de cambiar a otro sistema de base de datos completamente diferente a mitad de desarrollo, solo basta con configurar un archivo .yml. La capa de abstracción utilizada encapsula toda la lógica de los datos. El resto de la aplicación no tiene que preocuparse por las consultas SQL y el código SQL que se encarga del acceso a la base de datos.

2.8 Modelado de diseño

Antes de pasar a las particularidades del diseño, se hace necesario justificar por qué no se realiza el análisis. Como un elemento determinante se tiene el hecho de que las acciones de los usuarios que interactúan con el componente son muy sencillas y básicas aún para cualquier persona con pocos conocimientos de informática (dígase autenticarse, adjuntar un archivo para subirlo, descargar archivos, etc.), por lo que el proceso de captura de requisitos fue claro y explícito. Dentro de la metodología en uso, mediante el análisis los requisitos pueden ser refinados y estructurados para conseguir una mejor comprensión y estructuración del sistema (35), pero por lo antes expuesto, el desarrollador decidió ofrecer la visión general del sistema a través del estudio de los resultados del diseño y la implementación. El modelado del diseño crea un modelo de software enfocado en la representación de los datos, las funciones y el comportamiento requerido. Permite al ingeniero de software modelar el sistema o producto que se va a construir, posibilitando evaluar su calidad y efectuar mejoras antes de generar el código. (54)

REFERENCIAS BIBLIOGRÁFICAS

- [1] K. Fukunaga (1990). Introduction to Statistical Pattern Recognition. 2da. edición. Academic Press.
- [2] Enfoque Lógico Combinatorio al Reconocimiento de Patrones (I. Selección de variables y clasificación supervisada) – José Ruiz Shulcloper, Adolfo Guzmán Arenas, J. Francisco Martínez Trinidad.
- [3] Ruiz-Shulcloper, J., 2002. Logical Combinatorial Pattern Recognition: A Review. **Recent Research Developments in Pattern Recognition**, Volume 3, pp. 133-176.
- [4] Metodología de la Investigación - Daniel S. Behar Rivero.

Descripción del Problema del Dominio

El sistema presenta en su pantalla principal un mensaje de bienvenida a los usuarios junto con la opción para registrarse, o si ya se está registrado, poder entrar al sistema mediante la inserción del usuario/correo y una contraseña previamente escogidos, dichos datos deben validarse.

Una vez registrado el profesor, y después de haberse validado su acceso, puede seleccionar las siguientes actividades:

Subir Recurso: Permite a los profesores subir objetos de aprendizaje al sistema, para lo cual debe llenar algunos campos como son el título y la descripción.

Actualizar Recurso: Los profesores o los asesores a los estudiantes, podrán actualizar los recursos que han subido anteriormente, brindándoles la posibilidad de cambiarles los metadatos que los describen, cambiarles la posición que estos ocuparán dentro del entorno, así como borrarlos definitivamente de manera que estos sean los más recientes posibles, más fiables y mejor organizados para la mejor comprensión de los estudiantes.

Una vez que el estudiante entre al sistema, puede seleccionar las siguientes actividades:

Cargar BD: podrá cargar un archivo de base de datos en el formato CSV para obtener la matriz de aprendizaje(MA) pudiendo así visualizar los materiales ofrecidos por el profesor y los pasos a seguir como solución generada por el mismo sistema para la clasificación del problema según la base de datos cargada por el mismo estudiante, siendo este el paso principal para su inserción en la aplicación. El sistema tiene como objetivo que el estudiante domine las habilidades requeridas para el proceso de selección de los rasgos y los pasos a seguir para la clasificación del mismo cumpliendo así los objetivos de la asignatura Optativa Reconocimiento de Patrones.

Calcular MD:

Calcular MB:

Identificar TTs: identifica los testores típicos

Objeto de estudio

El objeto de estudio está encaminado en el proceso a seguir para la selección de rasgos utilizando la teoría de testores. desarrollar una aplicación informática para la selección de rasgos utilizando la teoría de testores, como apoyo al Proceso de Enseñanza y Aprendizaje (PEA) en la asignatura optativa Reconocimiento de Patrones.

Propuesta del Sistema

Con el objetivo de influir positivamente en el proceso de aprendizaje de la asignatura optativa Reconocimiento de Patrones en los estudiantes, quedó definido en el capítulo anterior el conjunto de herramientas a utilizar para el desarrollo de la aplicación. Se pretende desarrollar un sistema que garantice un eficiente desarrollo del conocimiento de los estudiantes con respecto al estudio

de selección de rasgos utilizando teoría de testores como apoyo a la enseñanza de asignatura optativa Reconocimiento de Patrones.

Indicadores y Descriptores de la dimensión Acceso a la Información:

Búsqueda de la información: El profesor usará fuentes confiables y pertinentes para encontrar la información más fiable y mejor organizada para la mejor comprensión en el proceso de aprendizaje para los estudiantes.

Selección de la información: El profesor elegirá los documentos digitales para resolver la necesidad de información.

Filtrado: (fecha de publicación, fuente reconocida sobre la temática y actualidad del contenido del artículo.