



INSTITUTO SUPERIOR MINERO METALURGICO
"Dr. Antonio Núñez Jiménez".
Facultad de Metalurgia - Electromecánica
Ingeniería Informática

Trabajo de Diploma

en opción al título de

Ingeniero Informático

Tema:

Mapa Social de los Investigadores del Centro de Estudio de la Energía y la Tecnología Avanzada de Moa (CEETAM) usando algoritmos de clustering.

Autor:

Yohandi Manuel López Llorente

Tutores:

Ms. C. Gustavo Rodríguez Bárcenas

Ing. Dabiel González Ramos

Moa, Cuba

Junio 2011

"Año 53 de la Revolución"

PENSAMIENTOS



“Los que no tienen el valor de sacrificarse han de tener, al menos, el pudor de callar ante los que se sacrifican”

José Martí

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al INSTITUTO SUPERIOR MINERO METALURGICO "Dr. Antonio Núñez Jiménez" para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del 2011.

Yohandi Manuel López Llorente
Nombre completo del primer autor

MSc. Gustavo Rodríguez Bárcenas
Nombre completo del primer tutor

Ing. Dabiel González Ramos
Nombre completo del segundo tutor

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado "Mapa Social de los Investigadores del Centro de Estudio de la Energía y la Tecnología Avanzada de Moa (CEETAM) usando algoritmos de clustering.", fue realizado en nuestra entidad: Instituto Superior Minero Metalúrgico de Moa. Se considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface:

___ Totalmente

___ Parcialmente en un ___ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

Como resultado de la implantación de este trabajo se reporta un efecto que económico asciende a _____ MN y _____ CUC.

Y para que así conste, se firma la presente a los ___ días del mes de _____ del año 2011.

Nombre del representante de la entidad Cargo

Firma

Cuño

DEDICATORIA

Este trabajo es para mis padres, y hermano por darme siempre ese indispensable apoyo para seguir adelante en la vida. Por creer siempre en mí y porque en las malas han sido mi regazo para desahogar mis penas, por haberme dado este tamaño y por apoyarme siempre en todo. A ti Ida, Pablo y Chino este regalo.

A mis tíos Irma, Ana, Marina, Marjoris, Israel, Tomás, Raúl, Eligio y Germán papá, que batallaron cual si fueran padres conmigo ustedes lo saben.

A mis abuelos, padres además. A Rey que seguro me observa orgulloso, a Mina, mi viejita querida, a Beby, y a mi abuela Monina, a ustedes también los quiero mucho.

A mi familia en general, y a mis primos, Yanniu, tú no me faltas, tampoco tú, Germán, eres mi otro hermano.

A Liumi, mi esposa, a Esmeraldo y Daysi por su apoyo por el ánimo que me brindaron.

A mis amigos y hermanos fieles y jodedores, Eduardo, Alexis José Manuel, a los amigos de la vida y el barrio, a todos ellos que siempre me soportan y están conmigo en las buenas y las malas.

En fin, a todos los que estuvieron ahí para mí cuando los necesité.

AGRADECIMIENTOS

Agradezco a mi familia por luchar firmemente a mi lado y ser ese pilar que me sostiene cuando invade el mal tiempo.

A mis amigos fieles, que forman otro fuerte pilar.

A mi primo Germán principalmente, por ayudarme en lo que hizo falta.

A todos los amigos de la vida, a mi gente de la primaria y secundaria, los de la vocacional, los de la UCI, los de acá del ISMM y los del barrio también, ustedes son los primeros.

A todos aquellos, sin excepción de ninguno, que han contribuido e influenciado mi vida para bien.

A todos ustedes por todo lo que hemos vivido, de veras Gracias...

RESUMEN

El surgimiento de nuevas tecnologías basadas en Web, han permitido que la comunicación e interacción entre usuarios sea cada vez más cercana. Este tipo de tecnología ha venido a cambiar la forma en que interactúan los usuarios, ya sea compartiendo archivos, el envío de mensajes, publicaciones en blogs, etc.

Debido al gran cúmulo de información que existe por esta se hace algo difícil encontrar el contenido que realmente se necesita.

En el Instituto Superior Minero Metalúrgico de Moa (ISMMM) cuenta con un sistema similar a las nuevas tecnologías mencionadas anteriormente, por ende cuenta con la misma situación de encontrar la información que realmente se necesita.

Específicamente en los portales dedicados a las publicaciones de artículos es que se pone vigente el uso de algoritmos y que aplicaremos en el sistema de del ISMM, uno de ellos son los algoritmos de **clustering o agrupamiento**, los cuales permiten estructurar y agrupar, características similares, en el sistema las relaciones que se producen entre los usuarios, en cuanto a características específicas definidas por los mismos fueron usadas para compilarlas usando los algoritmos de clustering, específicamente el de K-Medias. Esta utilización de técnicas de **clustering o agrupamiento**, ayudó a particionar un conjunto de datos en pequeños grupos que permiten un análisis eficiente de la información de los usuarios de nuestro sistema además de facilitarnos generar un gráfico con la interrelación investigativa de los investigadores usuarios del sistema antes mencionado.

ABSTRACT

The new Web-based technologies have allowed communication and interaction between users. They can share files, send messages, blog posts, etc.

Due to the large amount of information exists for this is so difficult to find the content they really need.

In the Mining Metallurgical Institute Moa (ISMMM) there is a similar system and it has the same situation to find the really needed information.

There are many sites on internet that use algorithms that we apply on ISMM's system, one of them is the clustering algorithms, they can to struct and to group a similar characteristics

Relationships between users on our system there are terms of specific characteristics defined by them and were used to compile using K-Means clustering algorithms. This technique helped to partition a dataset into small groups that allow an efficient analysis of information from users and to providing us generate a graph with the relationship between researching users on our system.

Tabla de Contenidos

| | |
|---|-----------|
| INTRODUCCION..... | 1 |
| CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA | 4 |
| INTRODUCCIÓN | 4 |
| 1.1 ESTADO DEL ARTE | 4 |
| 1.1.1 Antecedentes de los Algoritmos de Clustering..... | 4 |
| 1.1.2 Clustering: | 5 |
| 1.1.2.1 Concepto de clustering:..... | 5 |
| 1.1.2.2 Características de los algoritmos de clustering:..... | 6 |
| 1.1.3 Algoritmo de Clustering K-Media..... | 7 |
| 1.1.4 Modelo del Espacio Vectorial..... | 10 |
| 1.2 METODOLOGÍAS PARA EL DESARROLLO DE SOFTWARE | 10 |
| 1.2.1 El Proceso Unificado de Desarrollo (RUP). | 11 |
| 1.2.2 Metodología XP. | 12 |
| 1.2.2.1 Prácticas en las que se fundamenta la metodología XP. | 13 |
| 1.2.3 Metodología de desarrollo de software a usar en nuestro sistema. | 16 |
| 1.2 ARQUITECTURA DEL SISTEMA..... | 16 |
| 1.2.2 Arquitecturas en tres capas..... | 17 |
| 1.2.3 Patrón Arquitectónico Modelo Vista Controlador..... | 19 |
| 1.3.1 Ventajas del Modelo Vista Controlador | 20 |
| 1.4 TENDENCIAS Y TECNOLOGÍAS ACTUALES | 21 |
| 1.4.1 Lenguajes de programación..... | 21 |
| 1.4.1.1 Lenguajes del lado del cliente. | 22 |
| HTML | 22 |
| JAVA SCRIPT | 22 |
| 1.4.1.2 Lenguajes del lado del servidor. | 23 |
| PHP | 23 |
| 1.4.2 Frameworks de Desarrollo | 25 |
| 1.4.2.1 Microsoft .NET | 25 |
| 1.4.2.2 Symfony | 25 |
| 1.4.2.3 Codeigniter | 26 |
| 1.4.3 Sistemas Gestores de Bases de Datos. | 27 |
| 1.4.3.1 POSTGRES | 27 |
| 1.4.3.2 MySQL | 28 |
| 1.5 TECNOLOGÍAS UTILIZADAS | 29 |
| 1.5.1 PHP HTML y JAVASCRIPT Como Lenguaje de Programación..... | 29 |
| 1.5.2 MySQL Como Gestor de Base de Datos | 29 |
| 1.5.3 CodeIgniter Como FrameWork de Desarrollo | 30 |
| 1.6 HERRAMIENTAS UTILIZADAS | 31 |
| 1.6.1 XAMPP | 31 |
| 1.6.2 PhpDesigner 2008 v6.0.0..... | 32 |
| 1.7 CONCLUSIONES..... | 32 |
| CAPÍTULO 2: PLANEACIÓN Y DISEÑO..... | 33 |
| INTRODUCCIÓN..... | 33 |
| 2.1 PERSONAS RELACIONADAS CON EL SISTEMA. | 33 |
| 2.2 HISTORIAS DE USUARIOS | 34 |
| 2.2.1 Historias de Usuario definidas en el sistema..... | 34 |
| 2.3 PLANIFICACIÓN DE ENTREGAS..... | 35 |
| 2.3.1 Estimación de esfuerzos por historias de usuario..... | 36 |
| 2.3.2 Plan de entrega..... | 36 |
| 2.3.3 Plan de duración de la Iteración. | 36 |
| 2.4 LISTA DE RESERVA | 37 |

| | |
|--|-----------|
| 2.4.1 Requisitos Funcionales..... | 37 |
| 2.4.2 Requisitos No Funcionales..... | 37 |
| 2.5 TARJETAS CRC..... | 38 |
| 2.5.1 Módulo #1 Tratar y guardar la información insertada por el usuario..... | 38 |
| 2.6 CONCLUSIONES..... | 39 |
| CAPÍTULO 3: DESARROLLO Y PRUEBAS | 40 |
| INTRODUCCIÓN..... | 40 |
| 3.1 MODELACIÓN DE LOS DATOS..... | 40 |
| 3.2 DESARROLLO | 41 |
| 3.2.1 Tareas de ingeniería por historias de usuario..... | 42 |
| 3.2.1.1 Distribución de tareas por Historias de Usuario | 42 |
| 3.2.1.2 Tareas de la Historia de Usuario #1 Graficar Interrelación de los usuarios investigadores del CEETAM | 43 |
| 3.3 PRUEBAS..... | 43 |
| 3.3.1 Desarrollo dirigido por pruebas | 43 |
| 3.3.2 Pruebas de aceptación..... | 44 |
| 3.3.2.1 Prueba de Aceptación Validar Datos..... | 45 |
| 3.4 CONCLUSIONES..... | 45 |
| CAPÍTULO 4: ESTUDIO DE COSTO Y FACTIBILIDAD | 46 |
| 4.1 INTRODUCCIÓN..... | 46 |
| 4.2 EFECTOS ECONÓMICOS..... | 47 |
| 4.3 BENEFICIOS Y COSTOS INTANGIBLES EN EL PROYECTO | 48 |
| 4.4 FICHA DE COSTO | 48 |
| 4.5 ANÁLISIS DE RESULTADOS | 52 |
| 4.6 CONCLUSIONES..... | 54 |
| CONCLUSIONES GENERALES..... | 56 |
| RECOMENDACIONES..... | 57 |
| REFERENCIAS BIBLIOGRÁFICAS | 58 |
| BIBLIOGRAFÍAS | 61 |
| ANEXOS..... | 65 |
| Tarjeta CRC del modulo #2 Generar el gráfico general de los clusters..... | 65 |
| Tarjeta CRC del modulo #2 Generar el gráfico general de los clusters..... | 67 |
| Tareas de la Historia de Usuario #2 | 68 |
| Tareas de la Historia de Usuario #2 | 68 |
| Tareas de la Historia de Usuario #2 | 68 |
| Tareas de la Historia de Usuario #3 | 68 |
| Tareas de la Historia de Usuario #4 | 69 |
| Tareas de la Historia de Usuario #5 | 69 |
| Prueba de Aceptación Crear matriz de datos con el método espacio vectorial..... | 69 |
| Prueba de Aceptación cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de clustering K-Medias..... | 70 |
| Prueba de Aceptación Generar el gráfico general de los clusters | 70 |
| Prueba de Aceptación Mostrar distancias entre usuarios de un mismo cluster | 70 |
| Prueba de Aceptación Mostrar distancias clusters o grupos..... | 71 |
| Prueba de Aceptación Mostrar Perfil de Usuario | 71 |
| IMÁGENES DEL MÓDULO..... | 72 |

INDICE DE TABLAS

| | |
|---|----|
| Tabla 1 Personas relacionadas con el sistema | 33 |
| Tabla 2 Historias de usuarios que definen nuestro sistema..... | 34 |
| Tabla 3 Historia de usuario: Tratar y guardar la información insertada por el usuario..... | 34 |
| Tabla 4 Historia de usuario: Graficar la interrelación entre los usuarios investigadores del CEETAM..... | 35 |
| Tabla 5 Estimación de esfuerzos por historias de usuario..... | 36 |
| Tabla 6 Plan de entrega. | 36 |
| Tabla 7: Duración de las iteraciones..... | 36 |
| Tabla 8: Tabla de <i>releases</i> | 37 |
| Tabla 9: Tarjeta CRC#1..... | 39 |
| Tabla 10: Distribución de tareas por Historias de Usuario..... | 42 |
| Tabla 11: Tarea de programación..... | 43 |
| Tabla 12: Planilla Prueba de Aceptación..... | 44 |
| Tabla 13: Prueba de Aceptación Validar Datos..... | 45 |
| Tabla 14: Costo en Moneda Libremente Convertible. | 49 |
| Tabla 15: Costo en Moneda Nacional. | 50 |
| Tabla 16: Tarjeta CRC#2..... | 66 |
| Tabla 17: Tarjeta CRC#3..... | 67 |
| Tabla 18: Tarea de programación..... | 68 |
| Tabla 19: Tarea de programación..... | 68 |
| Tabla 20: Tarea de programación..... | 68 |
| Tabla 21: Tarea de programación..... | 69 |
| Tabla 22: Tarea de programación..... | 69 |
| Tabla 23: Tarea de programación..... | 69 |
| Tabla 24: Prueba de Aceptación Crear matriz de datos con el método espacio vectorial..... | 69 |
| Tabla 25: Prueba de Aceptación cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de clustering K-Medias..... | 70 |
| Tabla 26: Prueba de Aceptación Generar el gráfico general de los <i>clusters</i> | 70 |
| Tabla 27: Prueba de Aceptación Mostrar distancias entre usuarios de un mismo cluster..... | 70 |
| Tabla 28: Prueba de Aceptación Mostrar distancias entre clusters o grupos | 71 |
| Tabla 29: Prueba de Aceptación Mostrar Perfil de Usuario..... | 71 |

INDICE DE ILUSTRACIONES

| | |
|--|----|
| Ilustración 1 Matriz de Ejemplo para ejecución del Cluster. | 8 |
| Ilustración 2 Matriz de Ejemplo para ejecución del Cluster. | 8 |
| Ilustración 3 Matriz de Ejemplo para ejecución del Cluster. | 8 |
| Ilustración 4 Matriz de Ejemplo para ejecución del Cluster. | 9 |
| Ilustración 5 Matriz de Ejemplo para ejecución del Cluster. | 9 |
| Ilustración 6 Fases de la Metodología XP. | 13 |
| Ilustración 7 Representación de la arquitectura en tres capas. | 18 |
| Ilustración 8 MVC Modelo-Vista-Controlador. | 20 |
| Ilustración 9 Flujo de la Aplicación con CodeIgniter. | 27 |
| Ilustración 10 <i>Planilla de Tarjeta CRC</i> | 38 |
| Ilustración 11 Tablas de la Base de Datos que usará el sistema. | 41 |
| Ilustración 12 Evaluación del Sistema en función del tiempo. | 51 |
| Ilustración 13 Imagen General de los Clusters. | 72 |
| Ilustración 14 Distancias entre usuarios dentro de un Cluster. | 72 |
| Ilustración 15 Distancias entre Clusters. | 73 |
| Ilustración 16 Perfil de Usuario. | 73 |



INTRODUCCION

La tendencia de las personas para formar grupos y crear nuevas relaciones es fundamental en la estructura de la sociedad y la forma en que tales grupos se van desarrollando a través del tiempo y como este crecimiento afecta directamente el comportamiento de la estructura de la sociedad.

A partir del paso del tiempo con el desarrollo de las nuevas tecnologías y la aparición de la Internet estas relaciones que se fomentaron entre diferentes grupos de personas tomaron su espacio, esto se hizo posible a través de las tecnologías basadas en Web.

Las nuevas tecnologías basadas en Web, han permitido que la comunicación e interacción entre usuarios sea cada vez más cercana. Es por eso que hoy en día podemos encontrar diferentes servicios Web que permiten a los usuarios interactuar con personas en diferentes partes del mundo. Este tipo de tecnología ha venido a cambiar la forma en que interactúan los usuarios, ya sea compartiendo archivos, el envío de mensajes, publicaciones en *blogs*, etc.

Sistemas como *Flickr*, *Twitter*, *MySpace* y *Facebook*, han permitido retomar la teoría de redes sociales para poder crear nuevas aplicaciones en base a este tipo de sistemas, ya que permiten modelar redes del mundo real con la gran cantidad de información que poseen.

Con este alto crecimiento en la generación de información, debido principalmente a la automatización de procesos y a los avances en las capacidades de almacenamiento de información las personas almacenan gran cantidad de información representada como datos para posteriormente realizar un análisis y administración de estos.

El principal objetivo de interactuar con estos datos es clasificarlos en pequeños grupos que describan sus características principales, basándose en la similitud o diferencia entre ellos. Es imposible analizar directamente dicha cantidad de datos, por lo que comúnmente se recurre a la utilización de técnicas de *clustering* o agrupamiento que ayudan a particionar un conjunto de datos en pequeños grupos que permiten un análisis eficiente de la información. El análisis de grandes volúmenes de datos no sólo puede brindar información adicional, sino también conocimiento nuevo.

En el Instituto Minero Metalúrgico de Moa (ISMMM) existe un sistema Web desarrollado para El Centro de Estudio de la Energía y Tecnología de Avanzada (CEETAM) a la que pertenecen todos los profesores que se dediquen o estén inmersos en una determinada investigación relacionada con este contexto. Por medio de este sitio estos pueden publicar todo lo referente a su investigación compartiéndola con todos aquellos interesados. Con el paso del tiempo ha aumentado el número de participantes y con ello el volumen de almacenamiento de datos, por lo que se ha hecho un poco difícil acceder a determinadas informaciones que antes se realizaba de una forma más sencilla. Dadas las condiciones anteriores concluimos nuestro **problema científico** de esta manera:

¿Cómo implementar el algoritmo de *clustering* k-media que permita visualizar un mapa con la interrelación entre los usuarios investigadores del CEETAM?

Objeto de Estudio: Aplicación de los Algoritmos de *Clustering* o agrupamiento.

Campo de Acción: Algoritmo de *Clustering* K-Medias en el contexto Científico Tecnológico del CEETAM.

Hipótesis: Si se programa una aplicación que permitiera generar un gráfico o mapa social usando el algoritmo de clustering o agrupamiento será posible conocer la interrelación entre los investigadores actores en nuestro Campo de Acción.

Objetivo General: Implementar el algoritmo de *clustering* (agrupamiento) K-Media para generar un mapa que permita mostrar las interrelaciones entre los usuarios investigadores del CEETAM.

De acuerdo a esta propuesta se derivan los siguientes **Objetivos específicos**.

1. Determinar el estado del arte vinculado con los sistemas que usen los algoritmos de *clustering* y que sirvan de base y fundamento teórico de la investigación.
2. Realizar el análisis y diseño del módulo que posibilite mostrar los principales parámetros a tener en cuenta para llevar a cabo las fases posteriores.
3. Desarrollar e implementar el módulo sobre una aplicación Web para graficar las interrelaciones de los usuarios investigadores del CEETAM que ingresen en el sistema.



4. Realizar el estudio de factibilidad que permita mostrar los costos y beneficios del sistema y su desarrollo.
5. Validar la solución propuesta con las pruebas correspondientes en el capítulo de pruebas del sistema.

Para el desempeño de estas tareas se utilizarán los métodos teóricos de **Análisis y Síntesis** y **Análisis de Contenido** para procesar la información y elaborar conclusiones, el **Histórico-Lógico** para el estudio de la evolución del marco teórico de la investigación, además de **Métodos Valorativos** para el desarrollo e implementación del módulo, se han evaluado las diferentes alternativas de solución; valorando los aspectos técnicos, económicos y de mercado, así como su optimización, el **Método Sistémico** fue usado para el estudio de los rasgos fundamentales del sistema: componentes, medio, y estructura, utilizando para profundizar en el conocimiento de la estructura de los cluster por medios matemáticos, aplicados en la investigación.

El presente trabajo constará de 4 capítulos. El capítulo 1 abordará sobre los fundamentos teóricos y el estado del arte de nuestro objeto de estudio. El capítulo 2 presentará todo lo concerniente a la concepción del trabajo, así como de mostrar la planeación y diseño de nuestro módulo. En el capítulo 3 se hará alusión a todo lo relacionado con la implementación y las pruebas correspondientes para garantizar una mayor calidad al producto. En el cuarto capítulo se hará un análisis del costo y factibilidad correspondiente al producto creado y por último haremos el análisis de resultados. Además de las conclusiones y recomendaciones y las referencias y materiales bibliográficos consultados

1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se abordarán cuestiones generales concernientes a los patrones, el modelo espacio vectorial, los algoritmos de *clustering*, las tecnologías actuales donde se tratará lo referente a la metodología empleada para el desarrollo de la aplicación, los sistemas de gestión de bases de datos utilizado en la misma.

Se estudia la disponibilidad de aplicaciones dedicadas a este fin y conjuntamente se abordarán de forma general las metodologías de desarrollo de software, enfatizándose en los criterios para el desarrollo de la solución propuesta.

Se define además las herramientas que se usarán para el cumplimiento de los objetivos trazados, así como la tecnología a utilizar y el porqué de darle solución a estos a través de una aplicación Web.

1.1 Estado del Arte

1.1.1 Antecedentes de los Algoritmos de Clustering

Muchas son las ciencias que han participado en el desarrollo exitoso de herramientas con el fin de solucionar diversos problemas prácticos y teóricos. El reconocimiento de patrones es una zona relativamente joven de carácter interdisciplinario, encargada de diseñar o programar sistemas automáticos capaces de resolver problemas en diversas áreas. Con la llegada de las computadoras se crea un horizonte lleno de posibilidades, ya que además de almacenar gran cantidad de información, con ellas se puede diseñar

o programar sistemas automáticos para el reconocimiento de patrones de gran utilidad para muchas ciencias.

Según la definición de Watanabe, un patrón es una entidad a la que se le puede dar un nombre y esta representada por un conjunto de propiedades medibles y las relacionadas entre ellas (vector de características). [15]

Patrón es sinónimo de objeto y podemos encontrarlo en disímiles lugares, relacionados lógicamente con el vector de características.

Para la clasificación de patrones se necesita un proceso previo de análisis de los datos que se conoce como Técnicas de agrupamiento (*Clustering*).

En muchas áreas, se incrementa la necesidad de obtener información de un conjunto de datos. Esto requiere de análisis y clasificación de los datos con el objetivo de características comunes en individuos pertenecientes al mismo grupo.

Los algoritmos de *clustering* son estrategias encargadas de descubrir la estructura interna de los datos, pero cada uno de ellos supone condiciones en la base de datos que los hace eficientes para algunos tipos de datos e ineficientes para otros.

1.1.2 Clustering:

El *clustering* consiste en agrupar una colección dada de patrones no etiquetados en un conjunto de grupos. En este sentido, las etiquetas están asociadas con los grupos, pero las categorías se obtienen únicamente de las propiedades de los datos.

1.1.2.1 Concepto de *clustering*:

“El *clustering* representa la división de datos en grupos de objetos similares llamados *clusters*”. *Clustering* es también conocido como clasificación no supervisada, en donde no se tienen asignación de grupos a clases ya predefinidas, sino que los grupos se van creando de acuerdo a las características de los datos.

Los grupos o *clusters*, son un conjunto de objetos que comparten características similares y juegan un papel muy importante en la manera en como la gente analiza y describe el mundo que los rodea. De forma natural, el humano se encarga de dividir objetos en grupos (*clustering*) y asignar objetos particulares a dichos grupos (clasificación).



El *clustering* es una de las técnicas más útiles para descubrir conocimiento oculto en un conjunto de datos. En la actualidad el análisis de *clustering* en minería de datos ha jugado un rol muy importante en una amplia variedad de áreas tales como: reconocimiento de patrones, análisis de datos espaciales, procesamiento de imágenes, cómputo y multimedia, análisis médico, economía, bioinformática y biometría principalmente. Esto ha hecho posible que el análisis de *clustering* se considere como una de las mejores técnicas para obtener conocimiento y realizar exploraciones en los datos.

1.1.2.2 Características de los algoritmos de *clustering*:

Las características deseables de la mayoría de los algoritmos de *clustering* son las siguientes:

- ❖ *Escalabilidad*. La mayoría de los algoritmos de *clustering* trabajan de manera apropiada con un número pequeño de observaciones (hasta 200 aproximadamente), mientras que se necesita una gran escalabilidad para realizar agrupamiento de datos en bases con millones de observaciones.
- ❖ *Habilidad para trabajar con distintos tipos de atributos*. Muchos algoritmos se han diseñado para trabajar solo con datos numéricos, mientras que en una gran cantidad de ocasiones, es necesario trabajar con atributos asociados a tipos numéricos, binarios, discretos y alfanuméricos.
- ❖ *Descubrimiento de clusters con formas arbitrarias*. La mayoría de los algoritmos de *clustering* se basan en la distancia euclidiana, lo que tiende a encontrar clusters todos con forma (circular) y densidad similares. Es importante diseñar algoritmos que puedan establecer clusters de formas arbitrarias.
- ❖ *Requerimientos mínimos en el conocimiento del dominio para determinar los parámetros de entrada*. La herramienta no debería solicitarle al usuario que introduzca la cantidad de clases que quiere considerar, ya que dichos parámetros en muchas ocasiones no son fáciles de determinar, y esto haría que sea difícil controlar la calidad del algoritmo.
- ❖ *Habilidad para tratar con datos ruidosos*. La mayoría de las BD contienen datos con comportamiento extraño, datos faltantes, desconocidos o erróneos. Algunos



algoritmos de *clustering* son sensibles a tales datos y pueden derivarlos a clusters de baja calidad.

- ❖ *Insensibilidad al orden de las observaciones de entrada.* Algunos algoritmos son sensibles al orden en que se consideran las observaciones. Por ejemplo, para un mismo conjunto de datos, dependiendo del orden en que se analicen, los clusters devueltos pueden ser diferentes. Es importante entonces que el algoritmo sea insensible al orden de los datos, y que el conjunto de clusters devuelto sea siempre el mismo.
- ❖ *Alta dimensionalidad.* Una BD o DW (*DataWarehouse*) puede contener varias dimensiones o atributos, por lo que es bueno que un algoritmo de *clustering* pueda trabajar de manera eficiente y correcta no sólo en repositorios con pocos atributos, sino también en repositorios con un alto espacio dimensional, o gran cantidad de atributos.
- ❖ *Clustering basado en restricciones.* Es un gran desafío el agrupar los datos teniendo en cuenta no sólo el comportamiento, sino también que satisfagan ciertas restricciones.
- ❖ *Interpretación y uso.* Los usuarios esperan que los resultados del *clustering* sean comprensibles, fáciles de interpretar y de utilizar.

Con estas características, se busca diseñar algoritmos más flexibles que sean capaces de manipular una gran variedad de requerimientos de acuerdo a las necesidades de los usuarios.

1.1.3 Algoritmo de Clustering K-Media

Es un método que permite asignar a cada observación el cluster que se encuentra más próximo en términos del centroide (media). En general, la distancia empleada es la euclídea. [3]

Pasos:

1. Se toman de manera arbitraria k clusters iniciales.
2. Para el conjunto de observaciones, se vuelve a calcular las distancias a los centroides de los clusters y se reasignan a los que estén más próximos. Se vuelven a



recalcular los centroides de los k *clusters* después de las reasignaciones de los elementos.

3. Se repiten los dos pasos anteriores hasta que no se produzca ninguna reasignación, es decir, hasta que los elementos se estabilicen en algún grupo.

Usualmente, se especifican k centroides iniciales y se procede al paso (2) y, en la práctica, se observan la mayor parte de reasignaciones en las primeras iteraciones. [3]

Ejemplo: Supongamos dos variables x_1 y x_2 y 4 elementos: A, B, C, D. con la siguiente tabla de valores:

| | x_1 | x_2 |
|---|-------|-------|
| A | 5 | 3 |
| B | -1 | 1 |
| C | 1 | -2 |
| D | -3 | -2 |

Ilustración 1 Matriz de Ejemplo para ejecución del Cluster.

Se quiere dividir estos elementos en dos grupos ($k=2$).

De modo arbitrario, se dividen los elementos en dos clusters (AB) y (CD) y se calculan los centroides de los dos *clusters*. [3]

Cluster (AB):

| \bar{x}_1 | \bar{x}_2 |
|---------------------|---------------------|
| $\frac{5-1}{2} = 2$ | $\frac{3+1}{2} = 2$ |

Ilustración 2 Matriz de Ejemplo para ejecución del Cluster.

Cluster (CD):

| \bar{x}_1 | \bar{x}_2 |
|----------------------|-----------------------|
| $\frac{1-3}{2} = -1$ | $\frac{-2-2}{2} = -2$ |

Ilustración 3 Matriz de Ejemplo para ejecución del Cluster.

En el paso (2), calculamos las distancias euclídeas de cada observación al grupo de centroides y reasignamos cada una al grupo más próximo. Si alguna observación se

mueve de grupo, hay que volver a calcular los centroides de los grupos. Así, las distancias son: [3]

$$d^2(A, (AB)) = (5 - 2)^2 + (3 - 2)^2 = 10$$

$$d^2(A, (CD)) = (5 + 1)^2 + (3 + 2)^2 = 61$$

Como A está más próximo al *cluster* (AB) que al *cluster* (CD), no se reasigna.

Se hace lo mismo para el elemento B:

$$d^2(B, (AB)) = (-1 - 2)^2 + (1 - 2)^2 = 10$$

$$d^2(B, (CD)) = (-1 + 1)^2 + (1 + 2)^2 = 9$$

Por lo cual, el elemento B se reasigna al *cluster* (CD) dando lugar al *cluster* (BCD). A continuación, se vuelven a calcular los centroides:

| Cluster | \bar{x}_1 | \bar{x}_2 |
|---------|-------------|-------------|
| A | 5 | 3 |
| (BCD) | -1 | -1 |

Ilustración 4 Matriz de Ejemplo para ejecución del Cluster.

Nuevamente, se vuelven a calcular las distancias para cada observación para ver si se producen cambios con respecto a los nuevos centroides:

| | A | (BCD) |
|---|----|-------|
| A | 0 | 52 |
| B | 40 | 4 |
| C | 41 | 5 |
| D | 89 | 5 |

Ilustración 5 Matriz de Ejemplo para ejecución del Cluster.

Como no se producen cambios, entonces la solución para $k = 2$ *clusters* es: A y (BCD). Si se quiere comprobar la estabilidad de los grupos, es conveniente volver a correr el algoritmo con otros *clusters* iniciales (una nueva partición inicial). Una vez considerados los *clusters* finales, es conveniente interpretarlos; para ello, se pueden cruzar con otras variables categóricas o se pueden ordenar de modo que los objetos del primer *cluster* aparezcan al principio y los del último *cluster* al final. [3]

1.1.4 Modelo del Espacio Vectorial

Este modelo es el más utilizado en la actualidad en los Sistemas de Recuperación de Información (especialmente en la Web). Este modelo entiende que los documentos pueden expresarse en función de unos vectores que recogen la frecuencia de aparición de los términos en los documentos. Los términos que forman esa matriz serían términos no vacíos, es decir, dotados de algún significado a la hora de recuperar información y por otro lado, estarían almacenados en formato “*stemmed*” (reducidos los términos a una raíz común, tras un procedimiento de aislamiento de la base que agruparía en una misma entrada varios términos). [5]

Un documento se modela como un vector (o fila de una matriz de términos y documentos) en el que se indican las apariciones de cada término de la base de datos en ese documento. Normalmente se trabajan con pesos, que representan las importancias de los términos en el documento y en la colección. Si un término aparece mucho en un documento, se supone que es importante en ese documento aunque si aparece en muchos documentos, ese término no es útil para distinguir ningún documento del resto de la colección. Lo que se intenta en este modelo es medir cuánto ayuda un término a distinguir un documento o texto de los demás. [5]

1.2 Metodologías para el desarrollo de Software

Las piedras angulares del proceso de desarrollo del software son: el proyecto, las personas y el producto; siendo las características del cliente, el entorno de desarrollo y las condiciones del negocio, elementos que influyen en el proceso. Existe una estrecha relación entre personas, proyecto, producto y proceso. Estos términos son conocidos como las cuatro «P» en el desarrollo de software. El resultado final de un proyecto software es un producto, donde intervienen personas a través de un proceso de desarrollo de software que guía los esfuerzos de las personas implicadas en el proyecto.

La calidad en el desarrollo y mantenimiento del software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales dependen de los sistemas informáticos para su buen funcionamiento. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos



transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". [9]

Las metodologías existentes en la actualidad se dividen en dos grandes grupos atendiendo a sus características: las metodologías tradicionales (RUP, MSF) y las metodologías ágiles (XP, SCRUM). Las primeras están pensadas para el uso exhaustivo de documentación durante todo el ciclo del proyecto mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios, la confianza en las habilidades del equipo y al mantener una buena relación con el cliente. Teniendo en cuenta ambos enfoques damos paso al análisis de dos de las metodologías más usadas actualmente. [9]

1.2.1 El Proceso Unificado de Desarrollo (RUP).

Es un proceso para el desarrollo de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. Como tres características esenciales está dirigido por casos de uso: que orientan al proyecto a la importancia para el usuario y lo que se quiere, está centrado en la arquitectura: que relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini-proyectos donde los casos de uso y al arquitectura cumplen sus objetivos de manera depurada. RUP propone cuatro etapas para el desarrollo de un producto: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. Estas etapas revelan que para producir una versión del producto en desarrollo se emplean todas las actividades de ingeniería pero con diferente énfasis; en las primeras versiones se hace más énfasis en el modelado del negocio, requisitos, análisis y diseño; mientras en las posteriores el énfasis recae sobre las actividades de implementación, pruebas y despliegue. Además contempla flujos de trabajo de soporte que involucran actividades de planificación de recursos humanos tecnológicos y financieros. El Proceso Unificado de Desarrollo tiene 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. Flujos de trabajo: [10]

- ❖ Modelamiento del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.



- ❖ **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- ❖ **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- ❖ **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- ❖ **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
- ❖ **Instalación:** Produce *release* del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- ❖ **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- ❖ **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- ❖ **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización. [10]

Es una de las metodologías más generales y más usadas de las que existen en la actualidad, pues está pensada para adaptarse a cualquier proyecto. Constituye además, una propuesta de proceso para el desarrollo de software orientado a objeto, utilizando UML (Unified Model Language), para describir todo el proceso basándose en componentes. Este lenguaje es estándar, con él se puede visualizar, especificar, construir y documentar los artefactos de un sistema. [10]

1.2.2 Metodología XP.

La metodología XP o Extreme Programming es una de las variantes de las metodologías ágiles con más aceptación en la comunidad internacional de

desarrollo. Esta metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final o cliente, pues, es uno de los requisitos para llegar al éxito del proyecto. Consta de cuatro fases: Planificación, Diseño, Codificación y Pruebas.



Ilustración 6 Fases de la Metodología XP.

1.2.2.1 Prácticas en las que se fundamenta la metodología XP.

1. Planificación incremental

La Programación Extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de retroalimentación que permitan conocer con precisión dónde estamos. Como es lógico, la planificación es iterativa: un representante del negocio decide al comienzo de cada iteración qué características concretas se van a implementar. [4]

2. Programación en parejas

La metodología XP incluye, como una de sus prácticas estándar, la programación en parejas. Nadie programa en solitario, siempre hay dos personas delante del ordenador. El hecho de que todas las decisiones las tomen al menos dos personas proporciona un mecanismo de seguridad enormemente valioso. Es muy difícil que dos personas se salten tareas por descuido o negligencia. El código siempre está siendo revisado por otra persona. La revisión de código es el método más eficaz de conseguir código de calidad, algo corroborado por numerosos estudios, muchos de los cuáles son anteriores a la Programación Extrema. En contra de lo que pueda parecer, los dos



desarrolladores no hacen lo mismo: mientras el que tiene el teclado adopta un rol más táctico, el otro adopta un rol más estratégico.[4]

3. Refactorización

Uno de los objetivos de XP es mantener la curva de costes tan plana como sea posible, por lo que existen una serie de mecanismos destinados a mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez. A este proceso básico para mantener el código en buena forma se le llama refactorización. La refactorización no sólo sirve para mantener el código legible y sencillo: también se utiliza cuando resulta conveniente modificar código existente para hacer más fácil implementar nueva funcionalidad. [4]

4. Diseño simple

Otra práctica fundamental de la Programación Extrema es utilizar diseños tan simples como sea posible. El principio es "utilizar el diseño más sencillo que consiga que todo funcione". La metodología XP nos pide que no vivamos bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras. XP define un "diseño tan simple como sea posible" como aquél que pasa todos los *tests*, no contiene código duplicado, deja clara la intención de los programadores (enfatiza el qué, no el cómo) en cada línea de código y contiene el menor número posible de clases y métodos. [4]

5. Testing

La ejecución automatizada de tests es un elemento clave de la metodología XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests o pruebas de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. Él es la persona mejor cualificada para decidir cuál es la funcionalidad más valiosa. Un efecto lateral importante de los tests es que dan una gran seguridad a los desarrolladores: es posible llegar a hacer cambios más o menos importantes sin miedo a problemas inesperados, dado que proporcionan una red de seguridad. La existencia de tests hace el código muy maleable. [4]

6. Propiedad colectiva del código



XP aboga por la propiedad colectiva del código. En otras palabras, todo el mundo tiene autoridad para hacer cambios a cualquier código, y es responsable de ellos. Esto permite no tener que estar esperando a otros cuando todo lo que hace falta es algún pequeño cambio. [4]

7. Integración continua

En muchos casos la integración de código produce efectos laterales imprevistos, y en ocasiones la integración puede llegar a ser realmente traumática, cuando dejan de funcionar cosas por motivos desconocidos. La Programación Extrema hace que la integración sea permanente, con lo que todos los problemas se manifiestan de forma inmediata. [4]

8. Cliente en el equipo

Algunos de los problemas más graves en el desarrollo son los que se originan cuando el equipo de desarrollo toma decisiones de negocio críticas. Esto no debería ocurrir, pero a la hora de la verdad con frecuencia no se obtiene retroalimentación del cliente con la fluidez necesaria. La metodología XP intenta resolver este tipo de problemas integrando un representante del negocio dentro del equipo de desarrollo. Esta persona siempre está disponible para resolver dudas y para decidir qué se hace y qué no se hace en cada momento, en función de los intereses del negocio. [4]

9. Entregas pequeñas

Siguiendo la política de la metodología XP de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Éstas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor como para que resulten valiosas para el cliente. [4]

10. Estándares de codificación

Para conseguir que el código se encuentre en buen estado y que cualquier persona del equipo pueda modificar cualquier parte del código es imprescindible que el estilo de codificación sea consistente. Un estándar de codificación es necesario para soportar otras prácticas de XP. [4]



1.2.3 Metodología de desarrollo de software a usar en nuestro sistema.

XP y RUP son dos grandes metodologías que después de analizar sus principales características y los aspectos más sobresalientes de cada una de ellas, se ha determinado la implantación de XP, una metodología ligera, con menos requerimientos de documentación y planificación para el desarrollo de la aplicación. Si se escoge RUP traería grandes dificultades como son:

Disímiles de artefactos: El hecho de realizar varios artefactos y mantenerlos actualizados consume mucho tiempo.

El poco personal de desarrollo: Al ser solo una persona a cargo del desarrollo de la aplicación, este tomaría varios roles en cada etapa y sería muy complejo cumplir con las actividades de cada uno de ellos. [11]

En futuras versiones: Se piensa añadirle otras funcionalidades, destacando que se desea lograr una funcionalidad mínima del software lo mas pronto posible.[11]

Se necesita: La documentación mínima necesaria para el futuro soporte y mantenimiento del producto final.[11]

Requisitos cambiantes: Los cambios en un proceso de desarrollo son inevitables, al aparecer un nuevo requisito hace que se tenga que comenzar una nueva iteración para dar cumplimiento a su funcionalidad. Como el proyecto esta en plena investigación y en un ambiente de desarrollo sujeto a cambios repentinos se sugiere una gran adaptatividad y pronta respuesta, lo cual RUP no ofrece. [11]

Planificación inexistente: La planificación que se realiza en las fases iniciales está sujeta a muchas variaciones en dependencia de los cambios que se experimenten en los requisitos. Por tanto se hace muy difícil planificar actividades específicas si no se tiene claro que se debe hacer realmente.

Los inconvenientes planteados pueden ser eliminados con la utilización de la metodología XP. [11]

1.2 Arquitectura del Sistema

1.2.1 Arquitectura cliente servidor.



Esta arquitectura consiste básicamente en que un programa –el cliente informático – realiza peticiones a otro programa –el servidor – que le da respuesta.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de las responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre el cliente y el servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni necesariamente en un solo programa. El servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La **arquitectura cliente-servidor** sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico.

Ventajas.

- Centralización del control: los accesos, los recursos, la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- Escalabilidad: se puede aumentar la capacidad de los clientes y el servidor por separado.

El servidor del cliente es una arquitectura de red que separa al cliente (a menudo un uso que utiliza una interfaz gráfica de usuario) de un servidor. Cada caso del software del cliente puede enviar peticiones a un servidor. Los tipos específicos de servidores incluyen servidores Web, los servidores de archivo, los servidores del correo, entre otros. Mientras que sus prototipos varían algo, su arquitectura básica sigue siendo la misma. Aunque esta idea se aplica en una variedad de maneras, en diversas clases de usos, el ejemplo más fácil de visualizar es el uso actual de páginas Web en internet.

1.2.2 Arquitecturas en tres capas.

Arquitectura en capas: es donde se define como organizar el modelo de diseño a través de capas, que pueden estar físicamente distribuidas, lo que quiere decir que los componentes de una capa solo puede hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y

la organización de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son consistentes de ningún detalle o interfaz de las superiores.

La programación por capas es un estilo de programación en el que el objetivo primordial es separar la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación del usuario.

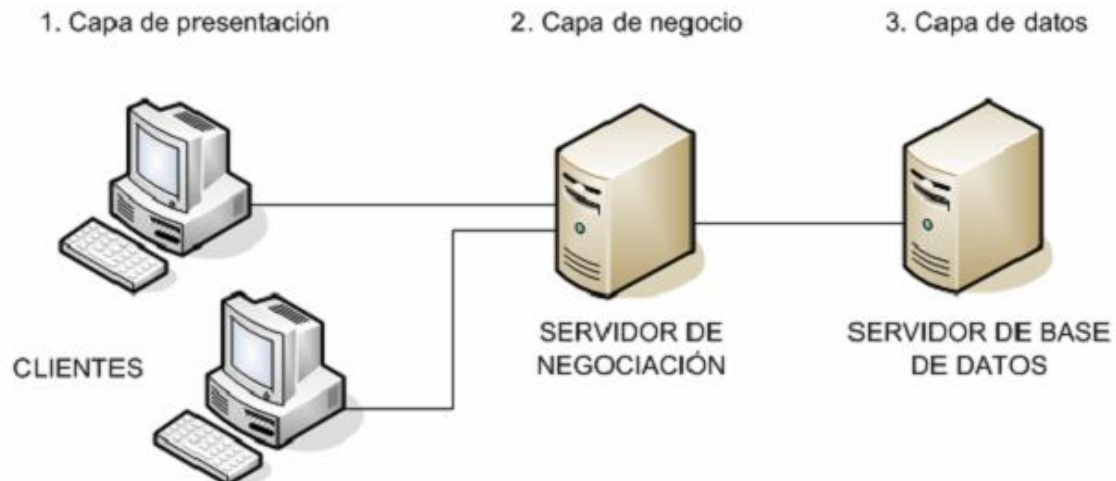


Ilustración 7 Representación de la arquitectura en tres capas.

Capas o niveles.

- ❖ Capa de presentación o interface: es la capa que le permite al usuario interactuar con el sistema, captura y le comunica la información al mismo, dando un mínimo de proceso (realiza un filtrado previo para asegurarse que no haya errores de formato). Esta capa se comunica únicamente con la de de negocio.
- ❖ Capa de negocio: es donde reside los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitarle al gestor de bases de datos almacenar o recuperar datos de él.
- ❖ Capa de datos: es donde residen los datos y es la encargada de acceder ellos. Está formada por uno o más gestores de bases de datos que se encargan de realizar el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



Todas estas capas pueden residir en un mismo ordenador aunque no es lo típico. Lo más usual es que haya una multitud de ordenadores donde reside la capa de interface (son los clientes de la arquitectura cliente/servidor). Las capas de negocios y de datos pueden residir en un mismo ordenador, y si el crecimiento de las necesidades lo aconseja, pueden dividirse en dos o más ordenadores. Así, si el tamaño o complejidad de las base de datos aumenta, pueden dividirse en varios ordenadores los cuales recibirán las peticiones del ordenador donde resida la capa de negocio. Si por el contrario, la complejidad fuese en la capa de negocio lo que obligase a la separación, esta lógica del negocio podría residir en uno o más ordenadores que realizarían las solicitudes a una única base de datos.

Ventajas:

- ❖ El estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual, permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales.
- ❖ El estilo admite muy naturalmente optimizaciones y refinamientos.
- ❖ Proporciona una amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa a medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la posibilidad de definir interfaces de capa estándar, a partir de las cuales pueden construirse extensiones o presentaciones específicas.

1.2.3 Patrón Arquitectónico Modelo Vista Controlador

La Arquitectura es el esqueleto o base de una aplicación. Representa la organización fundamental de un sistema. Desde los pequeños programas hasta los sistemas más grandes poseen una estructura y un comportamiento que los hace clasificables según su "arquitectura". En la Web es muy común la utilización de la arquitectura "3-capas", "n-capas", entre otras. El lenguaje de programación PHP con el *framework* de trabajo CodeIgniter, seleccionado anteriormente, implementa a su vez el patrón arquitectónico MVC, es por ello que adoptamos esta arquitectura para el desarrollo de la propuesta de solución. Modelo Vista Controlador (MVC). Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. [13]

- ❖ **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. [13]
- ❖ **Vista:** Presenta el modelo en un formato adecuado, como en una página Web que le permite al usuario interactuar con ella, usualmente un elemento de interfaz de usuario. [13]
- ❖ **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista. [13]

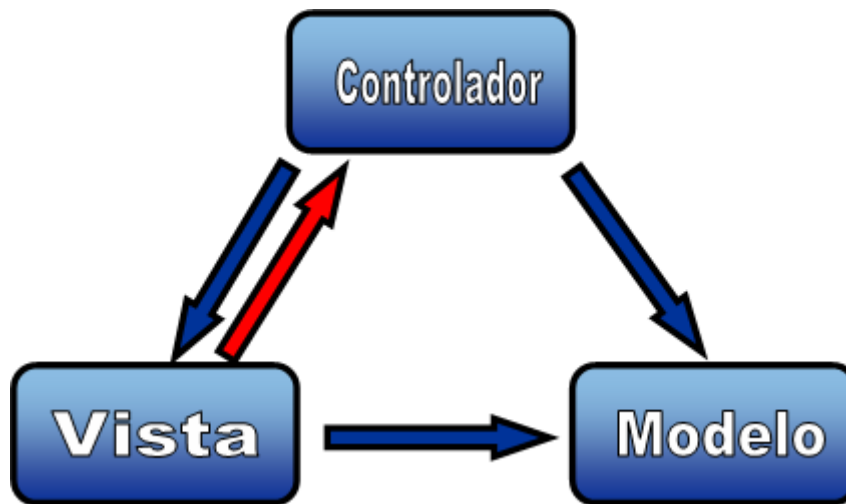


Ilustración 8 MVC Modelo-Vista-Controlador.

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) logrando un mantenimiento más rápido y sencillo de las aplicaciones. Ejemplo, para el caso de la Web, si se fuera a mostrar una misma aplicación en un navegador estándar, como en un navegador de un dispositivo móvil, sólo es necesario crear una vista nueva por cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (Aplicación de escritorio, HTTP, consola de comandos, *email*, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. [13]

1.3.1 Ventajas del Modelo Vista Controlador

- **Controlador:** Responde a eventos, usualmente acciones del usuario y realiza los cambios apropiados en el modelo o en la vista.



- La separación del Modelo de la Vista, es decir, separa los datos de la representación visual de los mismos.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en caso de errores.

Permite el escalamiento de la aplicación en caso de ser requerido. [13]

Teniendo lo antes expuesto usaremos el patrón arquitectónico MVC ya que el sistema al que se le desarrollará el módulo responde a las características del Modelo Vista Controlador

1.4 Tendencias y Tecnologías Actuales

1.4.1 Lenguajes de programación.

Uno de los ejes fundamentales que diferencian a Internet de otros medios de comunicación es la interacción y personalización de la información con el usuario. Esto se logra por medio de alguno de los diferentes lenguajes de programación para Web que existen en la actualidad. Dichos lenguajes se clasifican en dos partes fundamentales que reconocen la propia arquitectura Cliente/Servidor de esta plataforma de desarrollo: los lenguajes del lado del Servidor y los lenguajes del lado del Cliente.

Entre los lenguajes del lado del servidor podemos encontrar, entre los más sobresalientes por el auge que han tenido, algunos como PERL, ASP, PHP, Java, JSP, los módulos CGIs e ISAPIs, etc. Estos se caracterizan por desarrollar la lógica de negocio dentro del Servidor, además de ser los encargados del acceso a Bases de Datos y Tratamiento de la Información.

Del lado del cliente se encuentran principalmente el JavaScript, el Visual Basic Script y el HTML, los que se encargan de facilitar una interfaz así como de solicitar y mostrar las consultas y procedimientos necesarios a la programación lógica.

Esta distinción en los lenguajes ha sido necesaria debido a que la Web funciona en modo “Desconectado”, o sea, un usuario a través de un navegador hace una petición de una página Web a un Servidor Web (*Request*), el Servidor recepciona la petición, la procesa y le envía la Respuesta al Cliente (*Response*), este la recibe y se desconecta.



A continuación se abordan las características fundamentales de algunos de estos lenguajes.

1.4.1.1 Lenguajes del lado del cliente.

HTML

Lenguaje de Marcas de Hipertexto (*Hypertext Markup Language*) por sus siglas en inglés, es el lenguaje de marcado predominante para la construcción de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

HTML consiste de varios componentes vitales, incluyendo elementos y sus atributos, tipos de data, y la declaración de tipo de documento.

JAVA SCRIPT

Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del DOM.



Este lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores Web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas Web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Los autores inicialmente lo llamaron Mocha y más tarde Live Script pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995. JavaScript se puede incluir en cualquier documento HTML, o todo aquel que termine traducándose en HTML en el navegador del cliente; ya sea PHP, ASP, SVG... El código va inscrito dentro de los elementos HTML `<script>` y `</script>` se apoya su propuesta.

1.4.1.2 Lenguajes del lado del servidor.

PHP

PHP (*Personal Home Page*) es el acrónimo de *Hypertext Preprocessor*. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Es también un lenguaje interpretado y embebido en el HTML.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones Web actuales.

PHP es un lenguaje de programación de estilo clásico, esto quiere decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. No es un lenguaje de marcas como podría ser HTML, XML o WML.

A diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor, como por ejemplo podría ser, una base de datos. El programa PHP es ejecutado en el servidor y



el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

PHP es la gran tendencia en el mundo de Internet. Últimamente se puede observar un ascenso imparable, puesto que cada día son muchísimas más las páginas Web que lo utilizan para su funcionamiento, según las estadísticas, PHP se utiliza en más de 10 millones de páginas, y cada mes realiza un aumento del 15%. Como síntesis, PHP corre en 7 plataformas, funciona en 11 tipos de servidores, ofrece soporte sobre unas 20 Bases de Datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de aplicaciones web muy robustas, y contiene unas 40 extensiones estables sin contar las que se están experimentando, también tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros. Además de que:

Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.

Es muy rápido. Su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.

Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, resultará muy fácil aprender PHP.

Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP.

PHP tiene una de las comunidades más grandes en Internet, esto permite encontrar fácilmente ayuda, documentación, artículos, noticias, y otros recursos.

Permite las técnicas de Programación Orientada a Objetos.

No requiere definición de tipos de variables ni manejo detallado del bajo nivel.



1.4.2 Frameworks de Desarrollo

1.4.2.1 Microsoft .NET

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado. [6]

La plataforma .NET podría considerarse una respuesta de Microsoft al creciente mercado de los negocios en entornos Web, como competencia a la plataforma Java de Sun Microsystems y a los diversos *framework* de desarrollo Web basados en PHP. Su propuesta es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo. [6]

1.4.2.2 Symfony

Symfony es un completo *framework* diseñado para optimizar el desarrollo de las aplicaciones Web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación Web. [6]

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios Web de comercio electrónico de primer nivel. Symfony es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows. [6]



1.4.2.3 Codeigniter

Codeigniter es un poderoso *Framework* para PHP que facilita la escritura de código repetitivo, y a comparación de otros *Frameworks* como CakePHP, Symphony o Zend Framework.

Su principal objetivo es ayudar a que los desarrolladores, puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, a través de proveer un rico conjunto de librerías para tareas comúnmente necesarias, tanto como una simple interfase y estructura lógica para acceder a estas librerías. Codeigniter le permite concentrarse creativamente en su proyecto minimizando el volumen de código necesario para una tarea determinada. Hay que destacar que Codeigniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf (el creador de PHP) expresó que le gustaba Codeigniter "porque es rápido, ligero y parece poco un entorno". [6]

Codeigniter está basado en el patrón de desarrollo Modelo-Vista-Controlador. MVC es una aproximación al software que separa la lógica de la aplicación de la presentación. En la práctica, permite que sus páginas Web contengan mínima codificación ya que la presentación es separada del código PHP. Este *framework* tiene un enfoque bastante flexible del MVC, ya que los Modelos no son requeridos. Si no se necesita agregar separación, o se descubre que mantener los modelos requiera más complejidad de la que se quiere, puede ignorarse y construir la aplicación usando Controladores y Vista. También permite incorporar sus códigos existentes, o incluso desarrollar librerías de núcleo para el sistema, habilitándolo a trabajar en una forma que hace que tenga más sentido para el desarrollador. [6]

El siguiente gráfico ilustra como fluyen los datos a través del sistema:

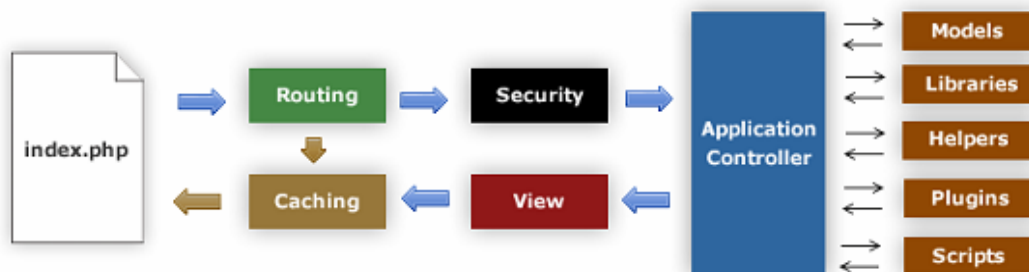


Ilustración 9 Flujo de la Aplicación con CodeIgniter.

1. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter.
2. El Router examina la petición HTTP para determinar que debe ser hecho con él.
3. Si un archivo de caché existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal.
4. Seguridad. Antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario es filtrado por seguridad.
5. El controlador carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica.
6. La Vista finalizada es presentada y enviada al explorador Web. Si el cacheo está habilitado, la vista es cacheada primero para que las peticiones subsecuentes puedan ser servidas. [6]

1.4.3 Sistemas Gestores de Bases de Datos.

Un Sistema de Bases de Datos (SBD) es una serie de recursos para manejar grandes volúmenes de información, sin embargo no todos los sistemas que manejan información son bases de datos. [6]

1.4.3.1 POSTGRES

Posee una estabilidad y confiabilidad legendaria nunca ha presentado caídas en varios años de operación de alta actividad. Está disponible 34 plataformas Unix en la última versión estable, existe una versión para Windows usando la plataforma Cygwin. Fue diseñado para ambientes de alto volumen intentando estar a la altura de Oracle, Sybase o Interbase. Escala muy bien al aumentar el número de CPUs y la cantidad de RAM.



Soporta transacciones y desde la versión 7.0, claves ajenas con comprobaciones de integridad referencial. Tiene mejor soporte para *subselects*, *triggers*, vistas y procedimientos almacenados en el servidor, además tiene ciertas características orientadas a objetos. Sin embargo consume muchos recursos y no escala bien en la plataforma Windows. Soporta un subconjunto de Sql92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos. Tiene límite del tamaño de cada fila de las tablas a 8k!!! (Se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento). [7]

1.4.3.2 MySQL

MySQL es un sistema de gestión de base de datos, multihilo y multiusuario con más de seis millones de instalaciones (Schumacher y Lentz, 2005). MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado lo ofrece bajo la GNU/GPL, pero, empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia que les permita ese uso. Está desarrollado en su mayor parte en ANSI C. [7]

Al contrario de proyectos como el Apache, donde el software es desarrollado unidad pública, y el copyright del código está en poder del autor individual, MySQL está poseído y patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius. [7]

MySQL funciona sobre múltiples plataformas, incluyendo AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2, Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista y otras versiones de Windows. También existe MySQL para OpenVMS (Kits of Open Source, 2005).

Posee características que son implementadas únicamente por este gestor de base de datos:



- ❖ Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- ❖ Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.[7]

1.5 Tecnologías Utilizadas

1.5.1 PHP HTML y JAVASCRIPT Como Lenguaje de Programación

Luego de hacer el análisis entre los lenguajes que implementan servicios Web, se decide utilizar el PHP embebido en el código HTML ya que:

- ❖ Está soportado en la mayoría de las plataformas de Sistemas Operativos, mientras que con ASP por ser propiedad de Microsoft no es multiplataforma.
- ❖ El PHP no tiene costo oculto, o sea que cuando se adquiere incluye un sin número de bibliotecas que proporcionan el soporte para la mayoría de las aplicaciones Web, por ejemplo e-mail, generación de ficheros PDF y otros. En caso de que no se tengan las bibliotecas están se pueden encontrar gratis en Internet. En el caso de ASP forma parte del *Internet Information Server* que viene integrado en Windows NT-2000 Server con su elevado costo de adquisición.
- ❖ PHP y ASP son parecidos en cuanto a la forma de utilización, pero PHP es más rápido, gratuito y multiplataforma.

Los lenguajes HTML y JavaScript para atender las peticiones del lado del cliente.[8]

1.5.2 MySQL Como Gestor de Base de Datos

Luego de analizadas las características y facilidades de los SGBD presentados y de la herramienta a desarrollar, se decide usar el MySQL como SGBD, por las siguientes razones:

- ❖ No se necesitará de un manejo complejo de la información.
- ❖ El PHP maneja más fácil al MySQL que a PostgreSQL, debido a la gran cantidad de funciones que tiene explícitas.



- ❖ El MySQL tiene mayor rendimiento y mayor velocidad mientras PostgreSQL es de 2 a 3 veces más lento que MySQL.
- ❖ Mejores utilidades de administración (backup, recuperación de errores, etc.).
- ❖ No suele perder información ni corromper los datos.
- ❖ No hay límites en el tamaño de los registros, PostgreSQL tiene un límite del tamaño de cada fila de las tablas a 8k.
- ❖ Mejor control de acceso, en el sentido de que usuarios tienen acceso a que tablas y con que permisos.
- ❖ MySQL se comporta mejor que PostgreSQL a la hora de modificar o añadir campos a una tabla "en caliente".
- ❖ MySQL por su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema, PostgreSQL consume más recursos y carga mas el sistema.
- ❖ El sistema al que se le integrara este módulo tiene su base de datos montada sobre MYSQL.[8]

1.5.3 CodeIgniter Como Framework de Desarrollo

Como *framework* de desarrollo se escoge CodeIgniter, puesto que:

- ❖ Se encuentra bajo la licencia *Open Source Apache/BSD-style*.
- ❖ Verdaderamente Liviano. El núcleo del sistema sólo requiere unas pocas pequeñas librerías. Esto es en duro contraste a muchos entornos de trabajo que requieren significativamente más recursos.
- ❖ Las librerías adicionales son cargadas dinámicamente a pedido, basado en sus necesidades para un proceso dado, así que el sistema base es muy delgado y bastante rápido.
- ❖ Usa el acercamiento Modelo-Vista-Controlador, que permite una buena separación entre lógica y presentación. Esto es particularmente bueno para proyecto en los cuales diseñadores están trabajando con sus archivos de plantilla, ya que el código en esos archivos será mínimo.



- ❖ Las URLs generadas por CodeIgniter son limpias y amigables a los motores de búsqueda. En vez de usar el acercamiento estándar "*query string*" a las URLs que es sinónimo de sistemas dinámicos, CodeIgniter usa un acercamiento basado en segmentos.
- ❖ Viene con un rango lleno de librerías que le permiten realizar las tareas de desarrollo Web más comúnmente necesarias, como acceder a una base de datos, mandar un email, validar datos de un formulario, mantener sesiones, manipular imágenes, trabajando con datos XML-RPC y mucho más.
- ❖ El sistema puede ser fácilmente extendido a través del uso de *plugins* y librerías asistentes, o a través de extensión de clases o ganchos del sistema. Aunque CodeIgniter si viene con un motor de plantillas simple que puede ser opcionalmente usado, no le fuerza a usarlo. Los motores de plantilla simplemente no pueden igualar el desempeño del nativo PHP, y la sintaxis que debe ser aprendida para usar un motor de plantilla es más fácil que aprender la base de PHP.[8]

1.6 Herramientas Utilizadas

1.6.1 XAMPP

XAMPP - Es un servidor independiente de plataforma, software libre, incluye el servidor Web Apache, los servidores de datos MySQL y SQLite, sus respectivos gestores phpMyAdmin y phpSQLiteAdmin, el intérprete del lenguaje homónimo PHP con los extras incluidos en Perl, el intérprete del lenguaje Perl, servidores de FTP como ProFTPD ó FileZilla FTP Server, las estadísticas Webalizer y OpenSSL, eAccelerator, Freetype2, libjpeg, libpng, zlib, Ming, etc. entre otros. El programa esta liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y MacOSX.

- ❖ Incluye Chequeo de seguridad.
- ❖ Contiene un Panel de control.

Incluye herramientas extras (Webalizer, Mercury Mail, Conmutador PHP). [1]



1.6.2 PhpDesigner 2008 v6.0.0

Es un completo entorno de desarrollo y programación especialmente diseñado para los gurús de PHP, aunque también permite trabajar con comodidad en otros lenguajes de programación como HTML, XHTML, CSS y SQL.

Ofrece toda una serie de asistentes y diálogos integrados que facilitan en todo de uso habitual, utilidades diversas y toda suerte de herramientas, todo ello en una interfaz de diseño sencillo y elegante que se puede personalizar con nada menos que dieciocho temas distintos.

Cuenta con cliente de FTP y navegador de ficheros integrado, utilidades de corrección y autocompletado, búsqueda integrada en google y soporte para proyectos, además de usar un práctico esquema de color para la sintaxis del código fuente que facilita enormemente la programación.

PhpDesigner soporta: PHP, HTML, XHTML, CSS, Java, Perl, JavaScript, VB, C# y SQL. [2]

1.7 Conclusiones

Partiendo del estudio del tema que nos interesa, los problemas y necesidades y de las características existentes, se brindan detalles de las herramientas y lenguajes que serán utilizados para la culminación de este trabajo, con los detalles anteriormente dados, las explicaciones de funcionalidad y los pasos a seguir para la automatización de esta herramienta podemos decir que está creada la base para la realización del presente trabajo de diploma.

2

CAPÍTULO 2: PLANEACIÓN Y DISEÑO

Introducción

En este capítulo se describen los parámetros principales que constituyen la entrada y salida de datos del algoritmo, que es el punto de partida para la construcción de las interfaces del mismo. Se presenta de la primera fase de la metodología, las historias de usuarios realizadas por el cliente, así como toda la planificación de entrega para su implementación. Se describen las funcionalidades que serán objeto de automatización mediante el empleo de las historias de usuarios (HU), se realiza una estimación del esfuerzo para las mismas y se establece un plan de iteraciones necesarias sobre el sistema para su terminación. Además se presentan las tarjetas Clases, Responsabilidades y Colaboradores, que permitirán trabajar con una metodología basada en objetos.

2.1 Personas relacionadas con el sistema.

| Persona(s) relacionada(s) con el sistema. | Justificación |
|--|---|
| Investigadores | Estas son las personas que estarán inmersos en las investigaciones que se desarrollen en el CEETAM, y están encargados de introducir los datos de las investigaciones, intereses entre otros. |
| Desarrollador | Es la persona encargada de llevar adelante la implementación del sistema |

Tabla 1 Personas relacionadas con el sistema

2.2 Historias de usuarios

La historia de usuario es la técnica utilizada en la metodología XP para la especificidad de los requisitos del software. Es el resultado directo del intercambio entre los usuarios y los desarrolladores a través de reuniones; mediante el diálogo y el intercambio de ideas en dichas reuniones se conocerán los requerimientos del sistema, las posibles soluciones y se definirán cuáles son las historias más importantes. En general se describen brevemente las características que el sistema debe tener desde la perspectiva del cliente.

| Historia de usuario | |
|--|--|
| Numero: (número de la historia de usuario) | Usuario: (usuario entrevistado para obtener los requisitos del sistema) |
| Nombre de la historia: (Nombre de la historia de usuario que sirve para identificarla mejor entre los desarrolladores y el cliente) | |
| Prioridad en negocios: (Importancia de la historia para el cliente) (Alta / Media / Baja) | Riesgo en desarrollo: (Dificultad para el programador) (Alto / Medio / Bajo) |
| Puntos estimados: 2 | Iteración asignada: (Iteración a la que corresponde) |
| Programador responsable: Nombre del Programador | |
| Descripción: (Se especifican las operaciones por parte del usuario y las respuestas que dará el sistema) | |
| Observaciones: (Algunas observaciones de interés, como glosario, información sobre usuarios, etc.) | |

Tabla 2 Historias de usuarios que definen nuestro sistema.

2.2.1 Historias de Usuario definidas en el sistema.

| Historia de usuario | |
|--|---|
| Numero: 1 | Usuario: <i>Gustavo Rodríguez Bárcenas</i> |
| Nombre de la historia: Tratar y guardar la información insertada por el usuario. | |
| Prioridad en negocios: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 4 | Iteración asignada: 1 |
| Programador responsable: <i>Yohandi Manuel López Llorente</i> | |
| Descripción: El usuario insertara los datos correspondientes y luego se tratará la información para el posterior uso. | |
| Observaciones: | |

Tabla 3 Historia de usuario: Tratar y guardar la información insertada por el usuario

| Historia de usuario | |
|---|--|
| Numero: 2 | Usuario: Gustavo Rodríguez Bárcenas |
| Nombre de la historia: Graficar la interrelación entre los usuarios investigadores del CEETAM | |
| Prioridad en negocios: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: 3 |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: El usuario solicitará el agrupamiento de los investigadores del CEETAM de manera general. | |
| Observaciones: Un mismo usuario puede realizar la utilización de la aplicación cuantas veces estime conveniente. | |

Tabla 4 Historia de usuario: Graficar la interrelación entre los usuarios investigadores del CEETAM.

2.3 Planificación de entregas

Para la plantación de la entrega se establece la prioridad de cada historia de usuario así como una estimación del esfuerzo necesario de cada una de ellas con el fin de determinar un cronograma de entregas.

Las estimaciones de esfuerzo asociado a la implementación de las historias se establecen utilizando como medida el punto. Un punto, equivale a una semana ideal de programación que en este caso sería de 5 días. Las historias generalmente valen de 1 a 5 puntos. Por otra parte, se mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración.

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

2.3.1 Estimación de esfuerzos por historias de usuario.

| Historia de usuario | Número | Puntos estimados |
|--|---------------|-------------------------|
| Tratar y guardar la información insertada por el usuario | 1 | 2 |
| Graficar la interrelación entre los usuarios investigadores del CEETAM | 2 | 5 |

Tabla 5 Estimación de esfuerzos por historias de usuario.

El plan de entregas se realiza teniendo en cuenta las unidades funcionales que se quieren entregar y cada uno de estos módulos abarca un número de historias de usuarios a implementar para dar cumplimiento al funcionamiento del mismo.

2.3.2 Plan de entrega.

| Módulos | Historia(s) de Usuario que abarca |
|--|--|
| Tratar y guardar la información insertada por el usuario | 1 |
| Creación de la matriz con el método espacio vectorial. | 2 |
| Cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de <i>clustering</i> K-Media. | 2 |
| Generar el gráfico general de los clusters. | 2 |

Tabla 6 Plan de entrega.

2.3.3 Plan de duración de la Iteración.

Teniendo en cuenta las Historias de usuarios anteriormente presentadas se realizarán una planificación en 5 iteraciones basándonos en el tiempo y procurando obtener la funcionalidad relacionada en la misma iteración.

| Iteración | Orden de implementación por Historias de Usuario | Duración total de la iteración en semanas |
|------------------|--|--|
| 1 | Tratar y guardar la información insertada por el usuario | 1 semana y 4 días |
| 2 | Capturar los datos necesarios para construir la matriz espacio vectorial. | 1 |
| 3 | Creación de la matriz con el método espacio vectorial. | 1 |
| 4 | Cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de <i>clustering</i> K-Media. | 3 |

Tabla 7: Duración de las iteraciones.

Combinando el plan de entrega y el plan de iteraciones se harán releases o liberaciones al sistema en las fechas mostradas a continuación:

| <i>Iteración \ Módulo</i> | <i>Generación de los módulos que tendrá el sistema</i> | <i>Mapa Social.</i> |
|---------------------------|--|---------------------|
| Fin 1ra iteración | 20 febrero 2011 | |
| Fin 2da iteración | 25 marzo 2011 | |
| Fin 3ra iteración | 26 abril 2011 | |
| Fin 4ta iteración | | 10 Junio 2011 |

Tabla 8: Tabla de *releases*.

2.4 Lista de Reserva

Después de conocer el personal involucrado con el sistema se procede a analizar las funcionalidades del sistema enumerándolas en una lista de reserva.

De acuerdo a lo antes expuesto nuestro sistema debe tener:

2.4.1 Requisitos Funcionales.

- ❖ Tratar y guardar la información insertada por el usuario.
- ❖ Extraer datos necesarios para construir la matriz Espacio-Vectorial.
- ❖ Calcular los clusters utilizando la matriz Espacio-Vectorial.
- ❖ Generar el gráfico general de los *clusters*.
- ❖ Mostrar las distancias entre usuarios de un mismo *cluster*.
- ❖ Mostrar distancias entre *clusters* o grupos.
- ❖ Mostrar el perfil del usuario seleccionado.

2.4.2 Requisitos No Funcionales.

- ❖ **Seguridad:** La aplicación debe ser capaz de tener un acceso limitado solamente a los usuarios del sistema.
- ❖ **Software:** El sistema debe contar con los programas necesarios para su buen comportamiento, entre estos encontramos un servidor que nos provea del gestor de base de datos seleccionado anteriormente y sea un intérprete homónimo del lenguaje PHP para su buen funcionamiento, recomendablemente XAMP, WAMP, además debe contar con un sistema operativo capaz de soportar con facilidad el servidor seleccionado.

Hardware: El computador que asumirá el rol de servidor debe ser una máquina capaz de responder a las exigencias de los requerimientos de software aunque la

computadora puede contar con un Processor: Intel Celeron, MMX, 433MHz una Memoria: 128 MB RAM.

- ❖ **Usabilidad:** El sistema será entendible por los usuarios, fácil de usar y manipular.

2.5 Tarjetas CRC.

En este epígrafe tiene lugar la realización de las tarjetas de clases, responsabilidades y colaboración, conocidas tradicionalmente como *tarjetas CRC*, las cuáles se realizan con el objetivo de facilitar la comunicación y documentar los resultados. Además, las mismas permiten la total participación y contribución del equipo de desarrollo en la tarea de diseño. Una tarjeta CRC representa un objeto, por tanto es una clase, cuyo nombre se ubica en forma de título en la parte superior de la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda y las clases implicadas con cada responsabilidad a la derecha, en la misma línea de su requerimiento correspondiente. Para mejor comprensión de las mismas, se determina agruparlas por HU. Esta nueva técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta brindando así la información que se necesita a la hora de implementar.

| Clase | |
|-------------------|----------------|
| Responsabilidades | Colaboraciones |
| Clase A | Clase B |

Ilustración 10 Planilla de Tarjeta CRC

2.5.1 Módulo #1 Tratar y guardar la información insertada por el usuario.

| StringManager | |
|--|---|
| Tratar y guardar la información insertada por el usuario | MySQLHandler listarPalabras(palabra) |



| | |
|--|---|
| | salvarPalabras(palabra) eliminarAsoicacion(palabras,usuario) asociarPalabras (palabras,usuario) getId (palabra) getStringEspecialidad (id) getCantidadAsoc(id_termino,id_usuario) eliminarAsociacionesString(string,usuario) FormacionComplementariaMgnt FormacionAcademicaMgnt InvestigacionesMgnt InteresesMgnt EspecialidadMgnt |
|--|---|

Tabla 9: Tarjeta CRC#1

Para continuar viendo las tarjetas CRC del sistema ver los [anexos](#).

2.6 Conclusiones

La Funcionalidades Iniciales del Sistema, la elección de las características a Agrupar, la Presentación del Mapa Social o gráfico de las interrelaciones de los usuarios del CEETAM, junto a las historias de usuarios creadas por el cliente constituyen el objeto de implementación de la presente investigación. A partir de las historias de usuarios se planifica la entrega del software, dicha entrega se realiza conjuntamente con el cliente que es el actor principal en la etapa de planificación de la metodología seleccionada. Se decide realizar el software en cuatro iteraciones de 1 semana y cuatro días 1, 1, 3 semanas de trabajo cada una para garantizar la entrega al cliente en el tiempo establecido, además de crearse 3 Tarjetas CRC, para una mejor concepción del sistema.

3

CAPÍTULO 3: Desarrollo y Pruebas

Introducción

En este capítulo se inicia la fase de desarrollo y pruebas según la metodología XP. Presentamos el modelo de datos utilizado para la aplicación concluyente, y se realiza el desarrollo de las iteraciones a partir del desglose de las HU en tareas.

3.1 Modelación de los Datos

La base de datos con que cuenta nuestro sistema, fue diseñada con anterioridad al desarrollarse el primer módulo para el CEETAM, a lo que solo se determinó que fuera necesario utilizar una tabla de dicha base de datos tabla (*users*) y crear luego dos entidades interrelacionadas con esta tabla, tabla (términos) y tabla (relación) para guardar las relaciones y términos abordados por los usuarios para su posterior procesamiento. Esta decisión mejora el rendimiento del sistema, pues en términos informáticos es menos eficiente y pesado procesar directamente la información extrayendo la información directamente de cadenas de caracteres. Dicho esto, el modelo de la base de datos concluye de esta manera:

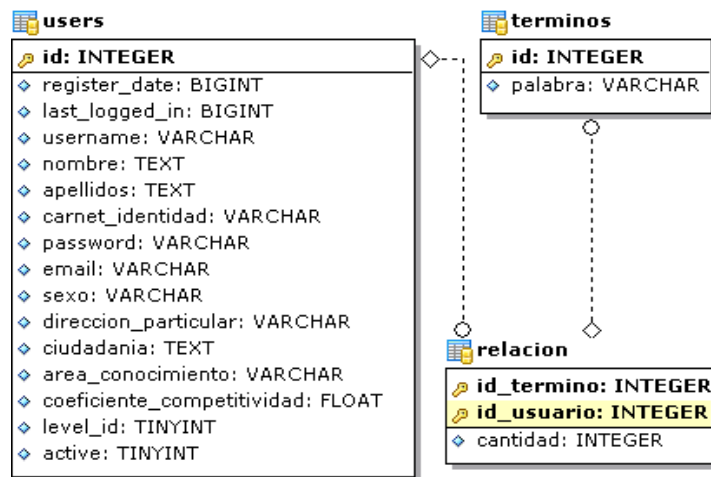


Ilustración 11 Tablas de la Base de Datos que usará el sistema.

3.2 Desarrollo

En la metodología XP el cliente es un integrante más del equipo de desarrollo pues crea las historias de usuario bajo la supervisión de los desarrolladores. Las historias se confeccionan cuando el cliente identifica con precisión lo que realmente desea, también debe estar presente cuando se realicen las pruebas de aceptación para cada historia de usuario, por lo que su presencia es imprescindible.

Cada historia de usuario puede o no dividirse en tareas de ingeniería (TI) o tareas de programación. Estas se crean para obtener una mejor planificación de la historia; con ellas se pretende cumplir con las funcionalidades básicas que luego conformarán las funcionalidades generales de cada historia.

Durante la fase planificación y diseño fueron detalladas las historias de usuario correspondientes a cada una de las iteraciones a desarrollar, teniendo en cuenta las prioridades y restricciones de tiempo, previstas por el cliente.

Para darle cumplimiento a cada HU, primero se debe realizar una revisión del plan de iteraciones, y si es necesario, se le hacen modificaciones a este.

Nuestro módulo inicialmente, debe ser capaz de tratar las cadenas insertadas al sistema para luego almacenarlas, este paso consiste en eliminar todos los conectores lógicos, las preposiciones, los artículos signos de puntuación entre otros a las cadenas de textos insertadas en los campos de Formación Académica, Formación Complementaria, Especialidad, Temática, Descriptor Temático y Palabras Claves de las Investigaciones y almacenar los datos restantes en la tabla términos de nuestra Base



de Datos donde cada usuario se relacionará con cada término citado por él y cada término contará con un contador que representará la cantidad de veces que está mencionado dicho término por el usuario que lo cita.

Con las veces que se cita cada término (contador), se ponderará una matriz, donde cada usuario representará un patrón y contará con una matriz o vector de características (Formación Académica, Formación Complementaria, Especialidad, Temática, Descriptor Temático y Palabras Claves de las Investigaciones); a esto podemos denominarlo espacio vectorial, pues nos permite realizar una serie de operaciones matriciales donde los resultados obtenidos quedarán dentro del conjunto donde se realiza la operación. A partir de aquí aplicamos el algoritmo de *clustering* K-media.

3.2.1 Tareas de ingeniería por historias de usuario.

Dentro del contenido del plan, las historias de usuario las historias de usuario se descomponen en tareas de programación o ingeniería y son asignadas al equipo de desarrollo para ser implementadas. Estas tareas necesariamente no tienen que ser entendidas por el cliente, pues solo serán usadas por los miembros del equipo de desarrollo, y pueden ser escritas en lenguaje técnico. Estas se pueden representar mediante las tarjetas de tareas.

3.2.1.1 Distribución de tareas por Historias de Usuario

| Historia de Usuarios | Tareas |
|---|---|
| Tratar y guardar la información insertada por el usuario | Tratar y guardar la información insertada por el usuario |
| Graficar Interrelación de los usuarios investigadores del CEETAM | Creación de la matriz con el método espacio vectorial. |
| | Cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de clustering K-Media. |
| | Generar el gráfico general de los clusters. |

Tabla 10: Distribución de tareas por Historias de Usuario

3.2.1.2 Tareas de la Historia de Usuario #1 Graficar Interrelación de los usuarios investigadores del CEETAM

| Tarea de Programación | |
|--|----------------------------|
| Número tarea: 1 | Número historia: 1 |
| Nombre tarea: Capturar los datos necesarios para construir la matriz espacio vectorial. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: | Fecha fin: |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: Aquí el usuario inserta los datos correspondientes y son guardados para luego procesarlos. | |

Tabla 11: Tarea de programación

Para ver más tareas por Historias de Usuario ver [anexos](#).

3.3 Pruebas

En la Programación Extrema es esencial el desarrollo de las pruebas, permitiendo probar continuamente el código. Cada vez que se desea implementar las funcionalidades que tendrá el software, XP propone una redacción sencilla de prueba, para ser pasada por el código posteriormente. El proceso constante de las pruebas permite la obtención de un producto robusto y con mayor calidad, lo que ofrece a los programadores una mayor certeza en el trabajo que desempeñan. En la metodología XP hay dos tipos de pruebas; las unitarias o desarrollo dirigido por pruebas, desarrolladas por los programadores verificando su código de forma automática, y las pruebas de aceptación, las cuáles son evaluadas luego de culminar una iteración, se verifica así, que se cumplió la funcionalidad requerida por el cliente. Con estas normas se obtiene un código simple y funcional de manera bastante rápida y eficiente. Por esto es importante pasar las pruebas al 100%.

3.3.1 Desarrollo dirigido por pruebas

El desarrollo dirigido por pruebas (DDP), se enfoca en la implementación orientada a pruebas. El código debe ser probado paso a paso para lograr un resultado, aunque no con lógica para el negocio, pero si funcional. Algunas personas confunden este término con las llamadas “pruebas de caja blanca” las cuáles se les practican a los métodos u operaciones para medir la funcionalidad del mismo, desde el punto de vista de validez del cliente. Sin embargo, el DDP se aplica antes de comenzar a implementar cada paso



de la tarea en desarrollo, asumiendo que la prueba es insatisfactoria desde un inicio. Sólo una vez que se haya cumplido de la forma más sencilla posible la lógica del código a probar se asume como cumplida. Luego se realiza un proceso conocido como “refactorización” de código perteneciente a una de las doce prácticas planteadas por la metodología XP, el cual consiste en mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez.

Es esencial el DDP y las pruebas de caja blanca se enfocan en la lógica del negocio.

3.3.2 Pruebas de aceptación

Las pruebas de aceptación en XP, se pueden asociar con las pruebas de caja negra que se aplican en otras metodologías de desarrollo, sólo que se crean a partir de las historias de usuario y no por un listado de requerimientos. Durante las iteraciones, las HU se traducen a pruebas de aceptación. En ellas se especifican desde la perspectiva del cliente, los escenarios para probar que la HU ha sido implementada correctamente. La misma puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo que persiguen estas pruebas, es garantizar que las funcionalidades solicitadas por el cliente han sido realizadas. Una HU no se considera completa hasta que no ha transitado por sus pruebas de aceptación. Luego de ver los paradigmas anteriores empleados para la realización de las pruebas y reunirse con el cliente para su análisis, el mismo decidió que se lleve a cabo el proceso mediante las pruebas de aceptación.

La planilla utilizada para plasmar el contenido de las pruebas de aceptación se muestra a continuación.

| Prueba de aceptación |
|---|
| HU: Nombre de la historia de usuario que va a comprobar su funcionamiento. |
| Nombre: Nombre del caso de prueba. |
| Descripción: Descripción del propósito de la prueba. |
| Condiciones de ejecución: Precondiciones para que la prueba se pueda realizar. |
| Entrada/Pasos de ejecución: Pasos para probar la funcionalidad. |
| Resultado esperado: Resultado que se desea de la prueba. |
| Evaluación de la prueba: Aceptada o Denegada. |

Tabla 12: Planilla Prueba de Aceptación.

3.3.2.1 Prueba de Aceptación Validar Datos.

| Prueba de aceptación |
|--|
| HU: Tratar y guardar la información insertada por el usuario. |
| Nombre: Validar datos. |
| Descripción: Guardar correctamente los datos en sus respectivas tablas para un buen funcionamiento del sistema. |
| Condiciones de ejecución: Deben llenarse los campos con los datos. |
| Entrada/Pasos de ejecución: Entrada de datos Eliminación de los conectores de las sentencias, conjunciones, preposiciones, artículos. Pronombres etc. Dejando las palabras claves. Guardar en la Base de Datos |
| Resultado esperado: Se almacenan correctamente los datos que se quieren. |
| Evaluación de la prueba: Aceptada. |

Tabla 13: Prueba de Aceptación Validar Datos

Para ver el resto de las pruebas de aceptación ver [anexos](#)

3.4 Conclusiones

Para lograr la completa implementación de cada historia de usuario en la fecha acordada con el cliente, estas se dividieron en tareas de ingeniería. A cada TI se le asignó un tiempo de desarrollo que se cumplió de manera eficiente garantizando así el objetivo principal de su confección.

Con la realización de las pruebas de aceptación el cliente se asegura de que las funciones implementadas cumplan su objetivo satisfactoriamente, probando individualmente cada módulo y asignándole la evaluación correspondiente. Todas las pruebas que se realizaron fueron aceptadas y el cliente estuvo conforme, cumpliendo con las historias de usuarios planteadas inicialmente.

4

CAPÍTULO 4: Estudio de Costo y Factibilidad

4.1 Introducción

Actualmente con el desarrollo de los proyectos viene adherido el estudio de factibilidad del sistema, imprescindible, pues se tiene en cuenta los costos a incurrir infiriéndose si el proyecto realizado será factible, o si no, no llevar a cabo el mismo.

Muchas son las formas de calcular el costo, pero para nuestro caso se utilizará la Metodología Costo Efectividad, la cual sugiere que la conveniencia de la ejecución de un proyecto se determina por la observación de ciertos factores en conjunto, estos son:

- ❖ El costo que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware / software y los costos de operación asociados.
- ❖ La efectividad que se entiende como capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo por el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad del cumplimiento del objetivo).

Esta parte es crucial en la elaboración de cualquier proyecto pues haciendo un estudio correcto de factibilidad se puede ahorrar meses e incluso años de trabajo, hasta evitar poner en duda la reputación profesional si se realiza un sistema mal planificado desde una etapa temprana.

4.2 Efectos Económicos

- ❖ Efectos directos.
- ❖ Efectos indirectos.
- ❖ Efectos externos.
- ❖ Intangibles.

Efectos directos:

POSITIVOS:

- ❖ Los usuarios autorizados a manipular el sistema tendrán la cobertura de ver como interrelacionarse con otros actores del sistema para enriquecer su investigación.
- ❖ Se facilitará el proceso de interrelacionar las investigaciones entre usuarios.

NEGATIVOS:

1. Debido a que nuestra aplicación está implementada con tecnología Web y además es un módulo de un sistema ya implementado con una serie de estándares de estilo y diseño se deberá usar el navegador Mozilla Firefox, pues los sistemas operativos de Windows instalados mayoritariamente cuentan con un navegador *IEExplore* que no soporta los estándares antes mencionados.

Efectos indirectos:

2. Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

Efectos externos:

3. Se tendrá una herramienta disponible que le facilitará gran parte de la interrelación investigativa entre los usuarios del CEETAM.

Intangibles:

4. En la valoración económica siempre hay elementos perceptibles por una comunidad como perjuicio o beneficio, pero al momento de ponderar en unidades monetarias esto resulta difícil o prácticamente imposible.



A fin de medir con precisión los efectos, deberán considerarse tres situaciones:

❖ **SITUACIÓN SIN PROYECTO**

La tendencia de crecer el número de usuarios del CEETAM hace que el proceso de compartir información se haga un poco engorroso debido a:

1. Mucha información sin referencias.

❖ **SITUACIÓN CON PROYECTO**

Para la entrada de los datos al sistema propuesto debemos seguir los siguientes pasos:

1. Tratar y guardar los datos a medir para su posterior visualización.
2. Generar el grafico final con las interrelaciones investigativas de los usuarios.

4.3 Beneficios y Costos Intangibles en el proyecto

Costos

- ❖ Resistencia al cambio.

Beneficios

- ❖ Mejor en cuanto comodidad para los usuarios.
- ❖ Mejora a calidad de información.
- ❖ Facilidad a la hora de interpretar la interrelación investigativa de los usuarios del CEETAM.

4.4 Ficha de costo

Para determinar el costo económico de nuestro proyecto se utilizará el procedimiento para elaborar Una Ficha de Costo de un Producto Informático. [14]

Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional.

Costos en Moneda Librementemente Convertible:

Costo en Moneda Librementemente Convertible.

| Ficha de Costo | |
|--|------------------|
| | Precio(s) |
| Costos Moneda Librementemente Convertible | |
| Costos Directos | |
| Compra de equipos de cómputo | 0,00 |
| Alquiler de equipos de cómputo | 0,00 |
| Compra de licencia de Software | 0,00 |
| Depreciación de equipos | 25,00 |
| Materiales directos | 0,00 |
| Subtotal | 25,00 |
| Costos Indirectos | |
| Formación del personal que elabora el proyecto | 0,00 |
| Gastos en llamadas telefónicas | 0,00 |
| Gastos para el mantenimiento del centro | 0,00 |
| Know How | 0,00 |
| Gastos en representación | 0,00 |
| Subtotal | 0,00 |
| Gastos de Distribución y Venta | |
| Participación en ferias o exposiciones | 0,00 |
| Gastos en transportación | 0,00 |
| Compra de materiales de propagandas | 0,00 |
| Subtotal | 0,00 |
| Total | 25,00 |

Tabla 14: Costo en Moneda Librementemente Convertible.

Costos en Moneda Nacional:

Costo en Moneda Nacional.

| Ficha de Costo. | |
|---|---------------|
| | Precio(s) |
| Costos Moneda Nacional | |
| Costos Directos | |
| Salario del personal que laborará en el proyecto | 100,00 |
| 5% del total de gastos por salarios se dedica a la seguridad social | 0,00 |
| 0.09% de salario total, por concepto de vacaciones a acumular | 0,00 |
| Gasto por consumo de energía eléctrica | 205,94 |
| Gastos en llamadas telefónicas | 0,00 |
| Gastos administrativos | 0,00 |
| Subtotal | 305,94 |
| Costos Indirectos | |
| Know How | 0,00 |
| Subtotal | |
| Total | 305,94 |

Tabla 15: Costo en Moneda Nacional.

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes, el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, tomaremos como costo el tiempo en horas empleado por los investigadores del CEETAM para tratar de relacionarse con otros investigadores en cuanto el contexto investigativo y la variable sería la complejidad de las pruebas que se realizan durante este proceso.

Solución A usando el sistema sin el módulo creado

1. El usuario accede al sistema y busca los usuarios del sistema que desee para ver su información (202 min.).

2. El usuario determina la compatibilidad en cuanto a las áreas de intereses insertados previamente y los intereses insertados por demás usuarios. (11 min.)
3. El usuario satisface sus intereses y determina en posible grupo de intereses investigativos estará. (25 min.)

Solución B, solución manual

1. El usuario solicita entrevista con cada investigador del CEETAM. (250 min.)
2. El usuario solicita a cada investigador del CEETAM los intereses investigativos, formación académica y complementaria además de su especialidad mediante las entrevistas. (500 min.)
3. Luego determina la compatibilidad en cuanto a los términos antes mencionados. (45 min.)

Valores de la variable (Solución con el módulo creado):

1. El usuario selecciona el vínculo que lo llevará a la representación gráfica de los grupos formados. (0.5 min.)
2. El usuario localizará en el mapa los grupos existente (2 min)
3. El usuario identificará a que grupo pertenece en cuanto a los campos mencionados anteriormente. (2 min.)

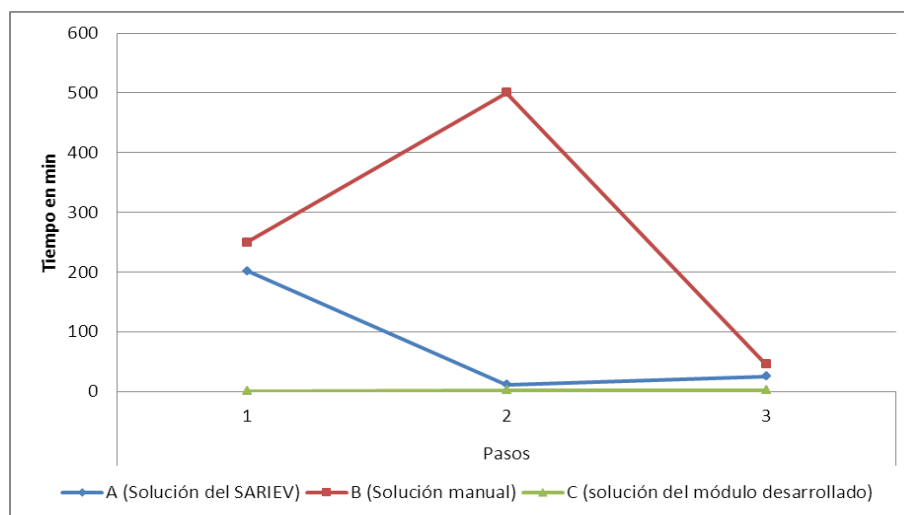


Ilustración 12 Evaluación del Sistema en función del tiempo.



Teniendo en cuenta los resultados reflejados en la gráfica en cuanto al Punto de Equilibrio queda demostrada la factibilidad del sistema evidenciado por la relación entre la complejidad del problema (cantidad de variables) y el tiempo que demora la solución del mismo de forma manual y automatizada.

4.5 Análisis de Resultados

Luego de la implementación y las pruebas realizadas al software se procede a realizar el análisis de resultados donde tras unas series de pruebas arrojaron los siguientes resultados sobre nuestro servidor, el cual cuenta con las siguientes características:

Servidor (PC Pentium III)

Sistema Operativo usado: Microsoft Windows 2003 Server Service Pack2

Processor: Intel(R) Pentium(R), Dual CPU 2.00 GHz

Memoria: 2 GB RAM.

Evaluación del módulo con datos no reales:

Se comienza a llenar la base de datos con usuarios creados no reales, de personas que no pertenecen al Centro de Estudio de Energía y Tecnología Avanzada en Moa. El algoritmo usado en la implementación de nuestro trabajo tiene complejidad orden n cuadrado y mediante su ejecución se obtienen los siguientes resultados:

- 1- La respuesta del sistema ocurre en unos pocos milisegundos.
- 2- El servidor se mantiene estable en sus procesos cada vez que se ejecuta la petición para mostrar los grupos creados.

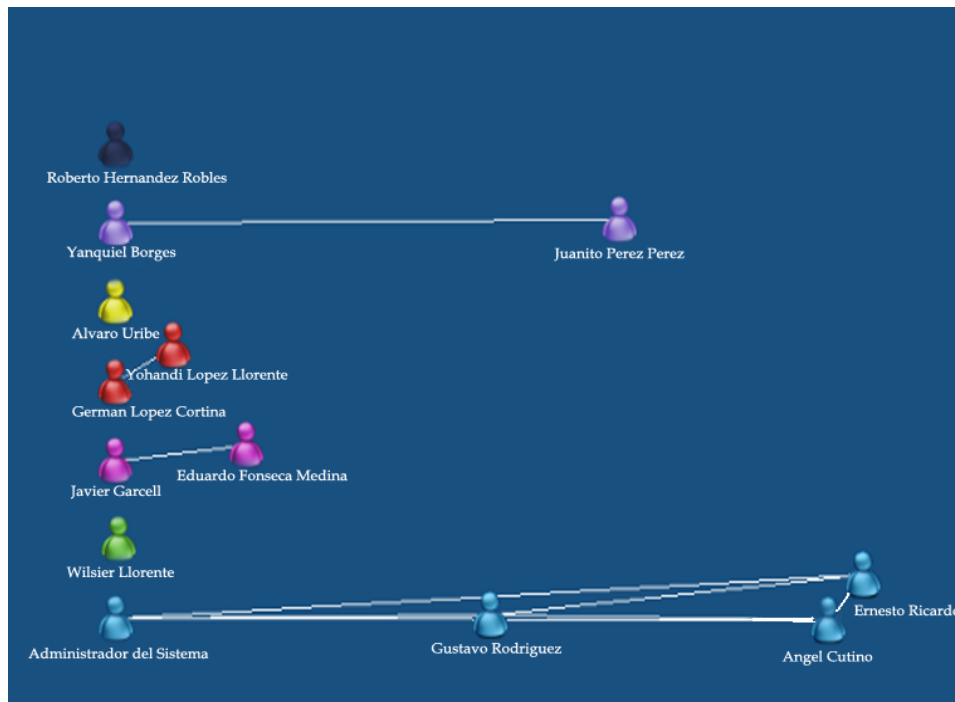


Ilustración 13 Grafico resultante que devuelve el sistema con usuarios no reales

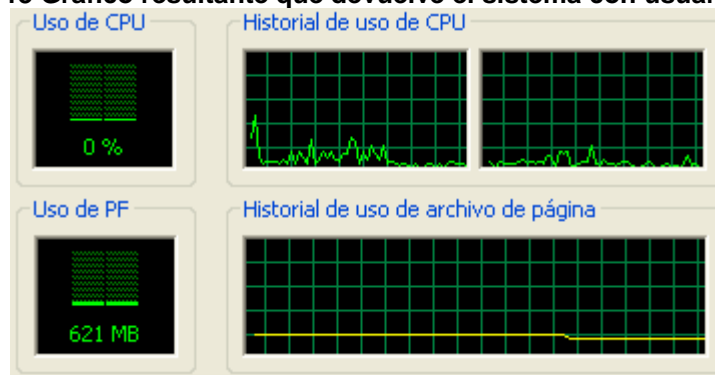


Ilustración 14 Comportamiento del uso del microprocesador y la memoria una vez activada la petición con los datos no reales.

Evaluación del módulo con datos reales:

Se comienza a llenar la base de datos con usuarios reales, personas que pertenecen al Centro de Estudio de Energía y Tecnología Avanzada en Moa donde se obtienen los siguientes resultados:

- 1- La respuesta del sistema continúa ocurriendo en unos pocos milisegundos.
- 2- El servidor se mantiene estable en sus procesos cada vez que se ejecuta la petición para mostrar los grupos creados.

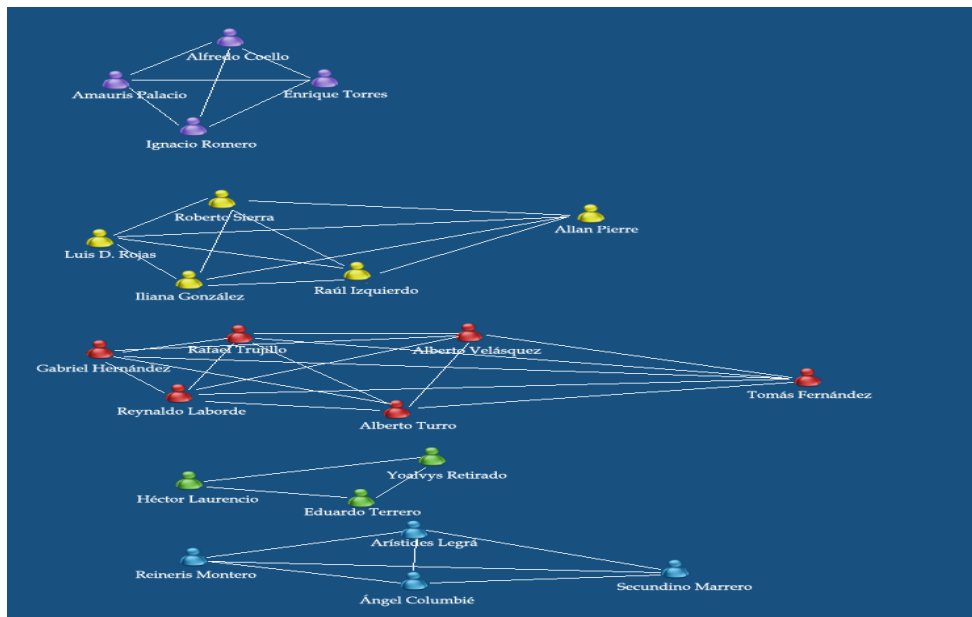


Ilustración 15 Grafico resultante que devuelve el sistema con usuarios reales.

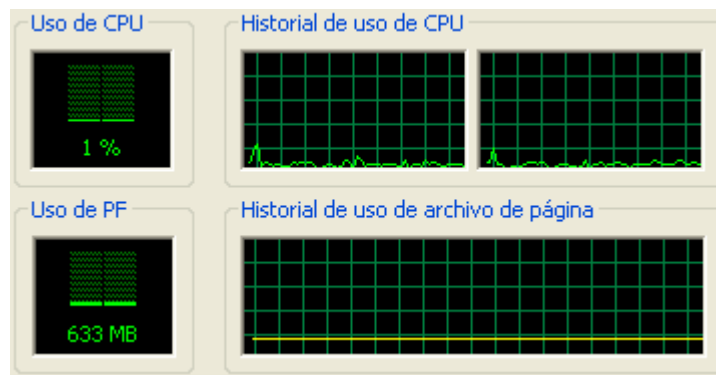


Ilustración 16 Comportamiento del uso del microprocesador y la memoria una vez activada la petición con los datos reales.

Teniendo lo antes expuesto el modulo implementado cumple con las expectativas del cliente donde a raíz de la implementación del algoritmo K-Media queda a disposición de los usuarios una herramienta que posibilitará la identificación de grupos a partir de su relación en cuanto áreas de conocimientos que domina, intereses entre otros.

4.6 Conclusiones

En este capítulo se hace un estudio profundo del costo real en que se incurrió durante el diseño e implementación del producto software mediante la Metodología Costo Efectividad (Beneficios), se analizaron todos los factores directos, indirectos, externos e intangibles, además se calculó el costo de ejecución del producto software mediante la



ficha de costo arrojando como resultados (**25,00 CUC y 305,94 CUP**) demostrándose la conveniencia de la elaboración del sistema. Además se le hizo al producto un análisis de resultados donde se le probaron con datos reales y no reales devolviendo en unos pocos milisegundos un gráfico con uno o más clusters (grupos) a los que podrían corresponder cada usuario registrado en el sistema demostrando la eficiencia del trabajo.



Conclusiones Generales

Realizada la investigación y luego de haber diseñado e implementado el software correspondiente arribamos a las siguientes conclusiones:

1. Se desarrolló un módulo sobre tecnología Web que permitirá mostrar la interrelación investigativa de los usuarios investigadores del CEETAM en el ISMMM.
2. Se llevó a cabo un estudio de las principales metodologías, lenguajes y herramientas que se consideraron factibles para el desarrollo del módulo, por lo cual se consideró XP como metodología, PHP como lenguaje de programación del lado del servidor, como servidor Web el Apache Server y como Gestor de Base de Datos el MySQL Server, tributando a las políticas de migración al Software Libre del ISMMM.
3. Se realizó la planificación y diseño de la aplicación, en los cuales se identificaron y especificaron los requerimientos funcionales, así como se llevaron a cabo las posteriores fases de implementación y prueba, que se definen en la metodología de desarrollo utilizada.
4. Se elaboraron las tarjetas CRC y se definió el uso del patrón de diseño Modelo Vista Controlador, también se desarrollaron las tareas correspondientes para dar solución a las historias de usuario.
5. Se realizaron las pruebas de aceptación lo que arrojó como resultado la conformidad del cliente, cumpliéndose con las historias de usuarios definidas en el Capítulo 3.
6. El estudio de factibilidad realizado siguiendo la metodología Costo Efectividad arrojó como resultado los efectos económicos y beneficios, así como el costo de ejecución del proyecto, siendo este 25,00 CUC. y \$ 305,94 MN demostrándose que es factible. También se le incluyó un análisis de resultados al proyecto demostrando la eficiencia del mismo.



RECOMENDACIONES

1. Realizar un estudio más profundo de este módulo en vista a perfeccionarlo en versiones futuras.
2. Ejecutar preferiblemente la aplicación en navegadores Open-Source debido que traducen los estilos de diseño con mayor claridad.



Referencias Bibliográficas

- [1] Aguilera, J. M. B; “*Sistema de Gestión Integral de la Empresa Empleadora del Níquel – Módulo de gestión de contratos de compras*”. Ing. Yodexy Mosqueda Naranjo. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 20-21 (2010).
- [2] Aguilera, J. M. B; “*Sistema de Gestión Integral de la Empresa Empleadora del Níquel – Módulo de gestión de contratos de compras*”. Ing. Yodexy Mosqueda Naranjo. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 41-42 (2010).
- [3] Anónimo; “Tema 5: Análisis de Cluster y Multidimensional Scaling”. Conferencia. Método de las k-medias, p. 13
- [4] Camejo, S. Y; “*openPlanner: Sistema para la gestión de planes de trabajo en el ISMM*”. Ing. Yodexy Mosqueda Naranjo. Meyquel García Avilés. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 29-32 (2010).
- [5] Cuza, E. R; “*Sistema Automatizado para la Recuperación de Información en Entornos Virtuales basado en Perfiles de Usuarios*”. Gustavo Rodríguez Barcenás. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 21-22 (2010).
- [6] Cuza, E. R; “*Sistema Automatizado para la Recuperación de Información en Entornos Virtuales basado en Perfiles de Usuarios*”. Gustavo Rodríguez Barcenás. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 41-43 (2010).
- [7] Cuza, E. R; “*Sistema Automatizado para la Recuperación de Información en Entornos Virtuales basado en Perfiles de Usuarios*”. Gustavo Rodríguez Barcenás. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 43-44 (2010).



- [8] Cuza, E. R; “*Sistema Automatizado para la Recuperación de Información en Entornos Virtuales basado en Perfiles de Usuarios*”. Gustavo Rodríguez Barcenás. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 50-51 (2010).
- [9] Muñoz, D. S; “*Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*”. Roiky Rodríguez Noa. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p 16-18(2009).
- [10] Muñoz, D. S; “*Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*”. Roiky Rodríguez Noa. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p 18-19(2009).
- [11] Muñoz, D. S; “*Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*”. Roiky Rodríguez Noa. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p 22-23(2009).
- [12] Muñoz, D. S; “*Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*”. Roiky Rodríguez Noa. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p 15-16(2009).
- [13] Muñoz, D. S; “*Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*”. Roiky Rodríguez Noa. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p 15-16(2009).
- [14] Pérez, A. M. C. *LÍNEA DE INVESTIGACIÓN: ESTUDIO DE USUARIOS*. Universidad Central de Las Villas, Cuba (2009).



[15] Romero, L. A, Cano, T. C “*Redes Neuronales y Reconocimiento de Patrones*”.
Universidad de Salamanca y Universidad de Valladolid. España.

Bibliografías

1. Aguilera, J. M. B; “*Sistema de Gestión Integral de la Empresa Empleadora del Níquel – Módulo de gestión de contratos de compras*”. Ing. Yodexy Mosqueda Naranjo Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 20-21 (2010).
2. Anónimo; “Tema 5: Análisis de Cluster y Multidimensional Scaling”. Conferencia. Método de las k-medias, p. 13
3. Camejo, S. Y; “*openPlanner: Sistema para la gestión de planes de trabajo en el ISMM*”. Ing. Yodexy Mosqueda Naranjo. Meyquel García Avilés. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 29-32 (2010).
4. Cuza, E. R; “*Sistema Automatizado para la Recuperación de Información en Entornos Virtuales basado en Perfiles de Usuarios*”. Gustavo Rodríguez Barcenás. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p. 21-22 (2010).
5. Muñoz, D. S; “*Desarrollo de una interfaz gráfica de usuario para el preprocesador meteorológico AERMET*”. Roiky Rodríguez Noa. Tesis en opción al título de ingeniero informático. Instituto Superior Minero Metalúrgico de Moa, Cuba. p 16-18(2009).
6. Pérez, A. M. G. *LÍNEA DE INVESTIGACIÓN: ESTUDIO DE USUARIOS*. Universidad Central de Las Villas, Cuba (2009).
7. Romero, L. A, Cano, T. C “*Redes Neuronales y Reconocimiento de Patrones*”. Universidad de Salamanca y Universidad de Valladolid. España.
8. Cambroner, C. G, Moreno, I. G “*Algoritmos de aprendizaje Knn & Kmeans, Inteligencia en Redes de Telecomunicación*”. Universidad Carlos III de Madrid.
9. MARÍN, A, BRANCH, John W. B “*Aplicación de dos nuevos algoritmos para agrupar resultados de búsquedas en sistemas de catálogos públicos en línea (OPAC)*”. Revista Interamericana de Bibliotecología. Ene.-Jun. 2008, vol. 31, no. 1, p. 47-65.

10. Baró, J. y Alemany, R. (2000): "Estadística II". Ed. Fundació per a la Universitat Oberta de Catalunya. Barcelona.
11. Peña Sánchez de Rivera, D. (1987): "Estadística. Modelos y Métodos. Volumen 2". Alianza Editorial. Madrid. ISBN: 84-206-8110-5
12. Johnson, R. R. (1996): "Elementary statistics". Belmont, etc. : Duxbury, cop
13. Martín-Guzmán, P. (1991): "Curso básico de estadística económica". AC, DL. Madrid. ISBN:84-7288-142-3
14. Baker JD, Bailey MK, Brahen NH et al. Selection of anesthesiology residents. Acad Med 1993; 68: 161-163
15. Pilon S, Tandberg D. Neural network and linear regression models in residency selection. AmJ Emerg Med 1997;15:361-364
16. Gunderman RB, Jackson VP. Are NBME examination scores useful in selecting radiology residency candidates? Acad Radiol 2000;7:603-606
17. Wagoner NE, Suriano JR, Stoner JA. Factors used by programs directors to select residents. J Med Educ 1986;61:10-21
18. Grantham JR. Radiology resident selection: results of a survey. Invest Radiol 1993;28:99-101
19. Smilen SW, Funai EF, Bianco HT. Residency selection: should interviewers be given applicants' board scores? Am J Obstet Gynecol 2001; 184:508-513
20. Baxt WG. Application of artificial neural networks to clinical medicine. Lancet 1995; 346:1135-1138
21. Sargent DJ. Comparison of artificial neural networks with other statistical approaches. Results from medical data sets. Cancer 2001;91:1636-1642
22. Freeman RV, Eagle KA, Bates ER et al. Comparison of artificial neural networks with logistic regression in prediction of in-hospital death after percutaneous transluminal coronary angioplasty. Am Heart J 2000;140:511-520



23. Steimann F. On the use and usefulness of fuzzy sets in medical AI. *Artificial Intelligence in Med* 2001; 21:131-137
24. Dayhoff JE, DeLeo JM. Artificial neural networks. Opening the black box. *Cancer* 2001; 91:1615-1635.
25. Adami, G., Avesani, P., Sona, D. Clustering documents in a web directory. *Proc. 5th ACM Int. Workshop on Web Information and Data Management*. New Orleans, USA. Pág. 66-73, 2003.
26. Aguilera, J., del Jesus, M., Herrera, F. Hibridación de Métodos Filtro y de Envoltura para Selección de Características. VIII Conf. de la Asoc. Esp. para la Inteligencia Artificial, Nov. 1999, Vol. I, 58-63.
27. Baeza-Yates, R., Poblete, B. A Website Mining Model Centered on User Queries. *European Web Mining Forum (EWMF 2005)*. Oporto, Portugal. 2005. Pág. 3-15.
28. Berry, M., Dumais, S., Shippy, A. A Case Study of Latent Semantic Indexing, Technical Report: UT-CS-95-271, University of Tennessee, USA, 1995.
29. Bezdek, J.C. *Pattern recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
30. Dhillon, I., Modha, D. Concept decompositions for large sparse text data using clustering, *Machine Learning*, 42(1), Pág. 143–175, 2001.
31. Duda, R., Hart, P. *Pattern Classification and Scene Analysis*. Wiley, 1973.
32. Gustafson, D.E., and Kessel, W., Fuzzy Clustering with a Fuzzy Covariance Matrix, in *Proc. IEEE-CDC*, Vol. 2 (K.S. Fu, Ed.), Pág. 761-766, IEEE Press, Piscataway, New Jersey, (1979).
33. Nettleton, D.F., Fuzzy covariance analysis, aggregation and input selection for fuzzy data. *IKBS '98. Int. Conf. on Knowledge Based Computer Systems*. Mumbai, India, Pág. 261-272, 1998.
34. Nettleton, D., Baeza-Yates, R. Web Retrieval: techniques for the aggregation and selection of queries and answers, (in Spanish), I Spanish Symposium



- on Fuzzy Logic and Soft Computing, Granada, Spain, Sept. 2005, Pág. 183-190.
35. Nettleton, D. Clustering and Aggregation of Web Query Session Data for User Profiling. Proceedings of Modeling Decisions in Artificial Intelligence, MDAI 2006, Tarragona, Spain, Abril 2006.
36. Zamir, O., Etzioni, O. Web document clustering: a feasibility demonstration. Proc. 21st annual int. ACM SIGIR conf. on Research and Development in Information Retrieval. Melbourne, Australia. Pág. 46 – 54, 1998.

Anexos

Historias de Usuario

| Historia de usuario | |
|---|--|
| Numero: 3 | Usuario: Gustavo Rodríguez Bárcenas |
| Nombre de la historia: Mostrar distancias entre usuarios de un mismo grupo o cluster | |
| Prioridad en negocios: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: 3 |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: El usuario podrá ver las distancias entre los usuarios investigadores dentro de un mismo cluster o grupo. | |
| Observaciones: Un mismo usuario puede realizar esta operación cuantas veces estime conveniente. | |

| Historia de usuario | |
|---|--|
| Numero: 4 | Usuario: Gustavo Rodríguez Bárcenas |
| Nombre de la historia: Mostrar distancias entre clusters o grupos formados. | |
| Prioridad en negocios: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: 3 |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: El usuario podrá ver las distancias entre los clusters o grupos a los que pertenecen los usuarios investigadores. | |
| Observaciones: Un mismo usuario puede realizar esta operación cuantas veces estime conveniente. | |

| Historia de usuario | |
|---|--|
| Numero: 5 | Usuario: Gustavo Rodríguez Bárcenas |
| Nombre de la historia: Mostrar perfil de usuario seleccionado. | |
| Prioridad en negocios: Alta | Riesgo en desarrollo: Alto |
| Puntos estimados: 5 | Iteración asignada: 3 |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: El usuario podrá ver el perfil del investigador que seleccione dentro de los grupos formados. | |
| Observaciones: Un mismo usuario puede realizar esta operación cuantas veces estime conveniente. | |

Tarjetas CRC

Tarjeta CRC del modulo #2 Generar el gráfico general de los clusters.

| StringManager | |
|----------------------|--------------|
| Tratar y guardar la | MySQLHandler |

| | |
|--|---|
| información insertada por el usuario. | listarPalabras(string) salvarPalabras(palabras) eliminarAsoicasion(palabras, usuario) asociarPalabras (palabras, usuario) getCantidadAsoc (id_termino, id_usuario) eliminarAsociacinoesString (string, usuario) |
| Creación de la matriz con el método espacio vectorial. | MySQLHandler getMatriz () |
| Cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de <i>clustering</i> K-Media. | Class Cluster masCercano (matriz,clusters) agrupamiento () clustering (matriz,clusters) ubicacion (usuario,clusters) buscarCluster (id,&clusters) Cluster (filas) calcularCentroide () distanciaRespecto (matriz) paraGraficar |
| Generar el gráfico general de los <i>clusters</i> | graficar () construirCoordenadasApartir(iniciox,inicioy,finy) nombresParaGraficar() tomarDatosMostrar () drawMe (&graft,iniciox,inicioy,image,finy) jpgraph_line jpgraph_scatter jpgraph |
| Mostrar distancias entre usuarios de un mismo grupo o cluster | Class Cluster NombresCompleto () buscar_nombre (valor) dame_nombre(valor) |
| Mostrar distancias entre clusters o grupos formados. | distancias_entre_grupos (listado) |
| Mostrar perfil de usuario. | datos_de_usuario (usuario) |

Tabla 16: Tarjeta CRC#2

Tarjeta CRC del modulo #2 Generar el gráfico general de los clusters

| Cluster | |
|--|--|
| Tratar y guardar la información insertada por el usuario. | listarPalabras(string) salvarPalabras(palabras) eliminarAsoicasion(palabras, usuario) asociarPalabras (palabras, usuario) getCantidadAsoc (id_termino, id_usuario) eliminarAsociacinoesString (string, usuario) |
| Creación de la matriz con el método espacio vectorial. | MySQLHandler getMatriz () |
| Cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de <i>clustering</i> K-Media. | Class StringManager calcularCentroide () adicionarFila (usuario, fila) eliminarFila (usuario) distanciaRespecto (matriz) contieneMatriz (usuario) |
| Generar el gráfico general de los <i>clusters</i> | getUsuarios () jpggraph_line jpggraph_scatter jpggraph |
| Mostrar distancias entre usuarios de un mismo grupo o cluster | NombresCompleto () buscar_nombre (valor) dame_nombre(valor) |
| Mostrar distancias entre clusters o grupos formados. | StringManager distancias_entre_grupos (listado) |
| Mostrar perfil de usuario. | StringManager datos_de_usuario (usuario) |

Tabla 17: Tarjeta CRC#3

Tareas de Ingeniería o Programación

Tareas de la Historia de Usuario #2

| Tarea de Programación | |
|--|---------------------|
| Número tarea: 2 | Número historia: 1 |
| Nombre tarea: Creación de la matriz con el método espacio vectorial. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 3 |
| Fecha inicio: | Fecha fin: |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: Los datos insertados son procesados con el método espacio vectorial obteniendo una matriz para un posterior cálculo | |

Tabla 18: Tarea de programación

Tareas de la Historia de Usuario #2

| Tarea de Programación | |
|---|---------------------|
| Número tarea: 3 | Número historia: 1 |
| Nombre tarea: Cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de <i>clustering</i> K-Media. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 4 |
| Fecha inicio: | Fecha fin: |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: Los datos o matriz obtenida en la interacción anterior es procesada mediante el algoritmo de clustering del K-Media. | |

Tabla 19: Tarea de programación

Tareas de la Historia de Usuario #2

| Tarea de Programación | |
|--|---------------------|
| Número tarea: 4 | Número historia: 1 |
| Nombre tarea: Generar el gráfico general de los <i>clusters</i> . | |
| Tipo de tarea: Desarrollo | Puntos estimados: 4 |
| Fecha inicio: | Fecha fin: |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: El sistema genera el grafico de los clusters formados de forma general. | |

Tabla 20: Tarea de programación

Tareas de la Historia de Usuario #3

| Tarea de Programación | |
|---|---------------------|
| Número tarea: 5 | Número historia: 1 |
| Nombre tarea: Mostrar distancias entre usuarios de un mismo grupo o cluster | |
| Tipo de tarea: Desarrollo | Puntos estimados: 3 |
| Fecha inicio: | Fecha fin: |

| |
|---|
| Programador responsable: Yohandi Manuel López Llorente |
| Descripción: Las distancias entre usuarios se mostrarán en un menú desplegable. |

Tabla 21: Tarea de programación

Tareas de la Historia de Usuario #4

| Tarea de Programación | |
|--|---------------------|
| Número tarea: 6 | Número historia: 1 |
| Nombre tarea: Mostrar distancias entre clusters o grupos formados. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 3 |
| Fecha inicio: | Fecha fin: |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: Las distancias entre los clusters o grupos se mostrarán en un menú desplegable. | |

Tabla 22: Tarea de programación

Tareas de la Historia de Usuario #5

| Tarea de Programación | |
|---|---------------------|
| Número tarea: 7 | Número historia: 1 |
| Nombre tarea: Mostrar perfil de usuario seleccionado. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 3 |
| Fecha inicio: | Fecha fin: |
| Programador responsable: Yohandi Manuel López Llorente | |
| Descripción: En el menú de las distancias entre los usuarios de un cluster se mostrarán los nombres de cada usuario involucrado en el cluster con un vínculo al perfil del mismo. | |

Tabla 23: Tarea de programación

Pruebas del Sistema

Prueba de Aceptación Crear matriz de datos con el método espacio vectorial.

| Prueba de aceptación |
|--|
| HU: Graficar Interrelación de los usuarios investigadores del CEETAM |
| Nombre: Creación de la matriz con el método espacio vectorial. |
| Descripción: Seleccionar de la tabla en la base de datos el valor que tiene cada palabra (esto se determina según la cantidad de veces que se en el sistema dicha palabra) y se pondera la matriz con este valor. |
| Condiciones de ejecución: El usuario que no tenga nada, no contará con dicha matriz. |
| Entrada/Pasos de ejecución: Selección de los datos requeridos. Creación de la matriz con todos los usuarios del CEETAM. |
| Resultado esperado: Se crea satisfactoriamente la matriz |
| Evaluación de la prueba: Aceptada. |

Tabla 24: Prueba de Aceptación Crear matriz de datos con el método espacio vectorial.

Prueba de Aceptación cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de clustering K-Medias.

| Prueba de aceptación |
|--|
| HU: Graficar Interrelación de los usuarios investigadores del CEETAM |
| Nombre: Cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de clustering K-Medias. |
| Descripción: Se van creando los clusters según el método k-medias usando la matriz creada anteriormente. |
| Condiciones de ejecución: El usuario que no tenga nada, aparecerá en clusters vacíos. |
| Entrada/Pasos de ejecución: Utilización de la matriz obtenida con el método espacio vectorial. Creación de los clusters con todos los usuarios del sistema. |
| Resultado esperado: Se crean satisfactoriamente los clusters. |
| Evaluación de la prueba: Aceptada. |

Tabla 25: Prueba de Aceptación cálculo de los clusters de los investigadores del CEETAM usando el algoritmo de clustering K-Medias

Prueba de Aceptación Generar el gráfico general de los clusters

| Prueba de aceptación |
|---|
| HU: Graficar Interrelación de los usuarios investigadores del CEETAM |
| Nombre: Generar el gráfico general de los clusters |
| Descripción: Se va generando el mapa según la cantidad de clusters que se crearon y las distancias existentes entre ellos. |
| Condiciones de ejecución: El usuario que este en un cluster vacío no será representado. |
| Entrada/pasos de ejecución: Utilización de los clusters obtenidos junto con la distancia asociada Generación del mapa usando estos datos |
| Resultado esperado: Se genera el mapa satisfactoriamente. |
| Evaluación de la prueba: Aceptada. |

Tabla 26: Prueba de Aceptación Generar el gráfico general de los clusters

Prueba de Aceptación Mostrar distancias entre usuarios de un mismo cluster

| Prueba de aceptación |
|--|
| HU: Mostrar distancias entre usuarios de un mismo grupo o cluster |
| Nombre: Mostrar distancias entre usuarios de un mismo cluster |
| Descripción: Se va creando un menú desplegable según la cantidad de clusters y con los usuarios que lo conforman dentro con sus respectivas distancias entre ellos. |
| Condiciones de ejecución: El usuario que este en un cluster vacío no será representado. |
| Entrada/pasos de ejecución: Utilización de los clusters obtenidos junto con la distancia asociada Generación del menú con las distancias. |
| Resultado esperado: Se crea el menú satisfactoriamente |
| Evaluación de la prueba: Aceptada. |

Tabla 27: Prueba de Aceptación Mostrar distancias entre usuarios de un mismo cluster

Prueba de Aceptación Mostrar distancias clusters o grupos

| Prueba de aceptación |
|--|
| HU: Mostrar distancias entre clusters o grupos formados. |
| Nombre: Mostrar distancias entre clusters o grupos |
| Descripción: Se va creando un menú desplegable según la cantidad de clusters con distancias correspondientes entre ellos. |
| Condiciones de ejecución: Los clusters vacíos no serán representado. |
| Entrada/pasos de ejecución: Utilización de los clusters obtenidos junto con la distancia asociada Generación del menú con las distancias. |
| Resultado esperado: Se crea el menú satisfactoriamente |
| Evaluación de la prueba: Aceptada. |

Tabla 28: Prueba de Aceptación Mostrar distancias entre clusters o grupos

Prueba de Aceptación Mostrar Perfil de Usuario

| Prueba de aceptación |
|---|
| HU: Mostrar Perfil de Usuario seleccionado |
| Nombre: Mostrar distancias entre clusters o grupos |
| Descripción: Se redirige al usuario al perfil del usuario seleccionado en el menú. |
| Condiciones de ejecución: |
| Entrada/pasos de ejecución: Datos guardados por el usuario en la creación de su perfil. |
| Resultado esperado: Se muestra la página con el perfil del usuario seleccionado |
| Evaluación de la prueba: Aceptada. |

Tabla 29: Prueba de Aceptación Mostrar Perfil de Usuario

IMÁGENES DEL MÓDULO

Inicio Mi Perfil Intereses Resúmenes Investigación Compatibilidad Recursos Compartidos Salir

Bienvenido(a):
 Área de conocimiento: Sin Definir Grado de Competitividad: SIN CALCULAR

Inicio
 Mi Perfil
 Detalles de Perfil
 Mis Intereses
 Mis Resúmenes
 Investigaciones
 Usuarios Compatibles
 Búsqueda por Intereses
 Búsqueda Manual
 Cambiar Contraseña
 Mostrar Grupos

Distancia entre los Usuarios de Grupos:

[Grupo :1](#)
[Grupo :2](#)
[Grupo :3](#)
[Grupo :4](#)
[Grupo :5](#)
[Grupo :6](#)
[Grupo :7](#)

Distancia entre Grupos:
[Ver Distancias entre Grupos:](#)

Ilustración 17 Imagen General de los Clusters.

Mis Intereses
 Mis Resúmenes
 Investigaciones
 Usuarios Compatibles
 Búsqueda por Intereses
 Búsqueda Manual
 Cambiar Contraseña
 Mostrar Grupos

Distancia entre los Usuarios de Grupos:

[Grupo :1](#)
 La Distancia entre el Usuario: [Administrador Sistema](#) y el Usuario [Ernesto Ricardo](#) es de 0
 La Distancia entre el Usuario: [Administrador Sistema](#) y el Usuario [Angel Cutino](#) es de 0
 La Distancia entre el Usuario: [Administrador Sistema](#) y el Usuario [Gustavo Rodriguez](#) es de 0
 La Distancia entre el Usuario: [Ernesto Ricardo](#) y el Usuario [Angel Cutino](#) es de 0
 La Distancia entre el Usuario: [Ernesto Ricardo](#) y el Usuario [Gustavo Rodriguez](#) es de 0
 La Distancia entre el Usuario: [Angel Cutino](#) y el Usuario [Gustavo Rodriguez](#) es de 0
[Grupo :2](#)
[Grupo :3](#)
[Grupo :4](#)
[Grupo :5](#)
[Grupo :6](#)
[Grupo :7](#)

Distancia entre Grupos:
[Ver Distancias entre Grupos:](#)

Ilustración 18 Distancias entre usuarios dentro de un Cluster.

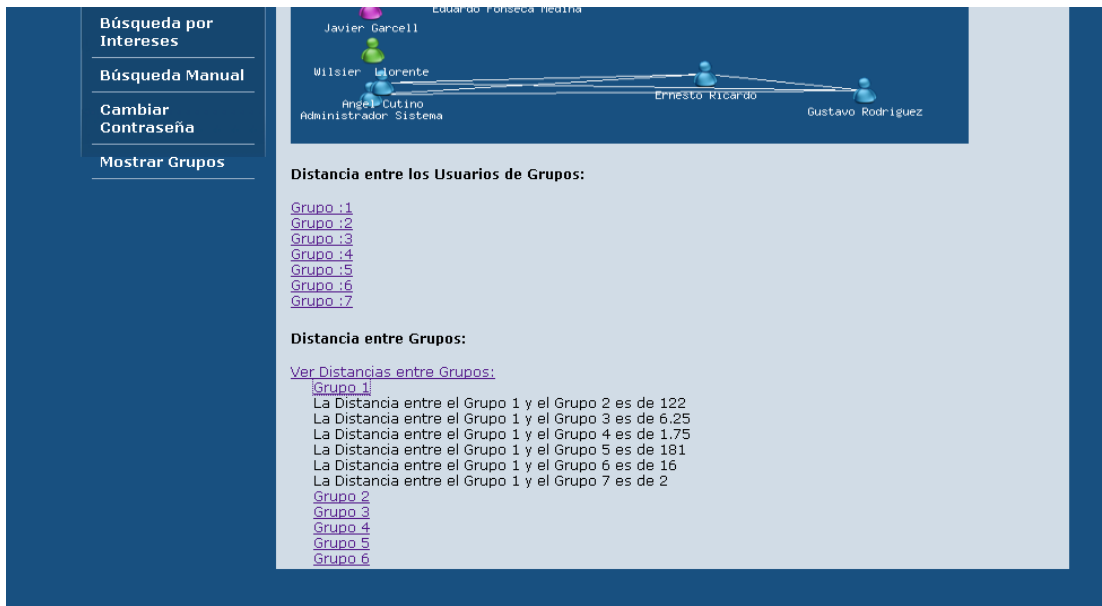


Ilustración 19 Distancias entre Clusters.

The screenshot shows a user profile page. At the top right, it says "Bienvenido(a): Yohandi Lopez LLorente (yohandi)" and "Área de conocimiento: Sin Definir Grado de Competitividad: SIN CALCULAR". The main heading is "Propiedades del Usuario". Below this, it identifies the user as "Administrador Sistema" from the "MINISTERIO DE EDUCACIÓN SUPERIOR INSTITUTO SUPERIOR MINERO METALÚRGICO DR. ANTONIO NÚÑEZ JIMÉNEZ". A profile picture placeholder is shown. The "DATOS PERSONALES" section includes: Nombre: Administrador, Apellidos: Sistema, Carnet de Identidad, E-mail: webmaster@ismm.edu.cu, Sexo: Masculino, Ciudadania: Cubana, and Dirección Particular: Edificio 8 Apartamento 27 Reparto Atlántico Moa-Holguin Cuba. The "Datos Docentes" section includes: Profesión Actual: Sin Definir, Grado Científico: Sin Definir, and Categoría Docente: Sin Definir.

Bienvenido(a): Yohandi Lopez LLorente (yohandi)
Área de conocimiento: Sin Definir **Grado de Competitividad:** SIN CALCULAR

Propiedades del Usuario

Administrador Sistema

MINISTERIO DE EDUCACIÓN SUPERIOR
 INSTITUTO SUPERIOR MINERO METALÚRGICO
 DR. ANTONIO NÚÑEZ JIMÉNEZ

DATOS PERSONALES

Nombre: Administrador **Apellidos:** Sistema
Carnet de Identidad: **E-mail:** webmaster@ismm.edu.cu
Sexo: Masculino **Ciudadania:** Cubana
Dirección Particular: Edificio 8 Apartamento 27 Reparto Atlántico Moa-Holguin Cuba

Datos Docentes

Profesión Actual: Sin Definir **Grado Científico:** Sin Definir
Categoría Docente: Sin Definir

Ilustración 20 Perfil de Usuario