

Universidad de las Ciencias Informáticas

Facultad 4



Título: Propuesta de un Sistema de Análisis de la Información Académica para Toma de Decisiones.

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

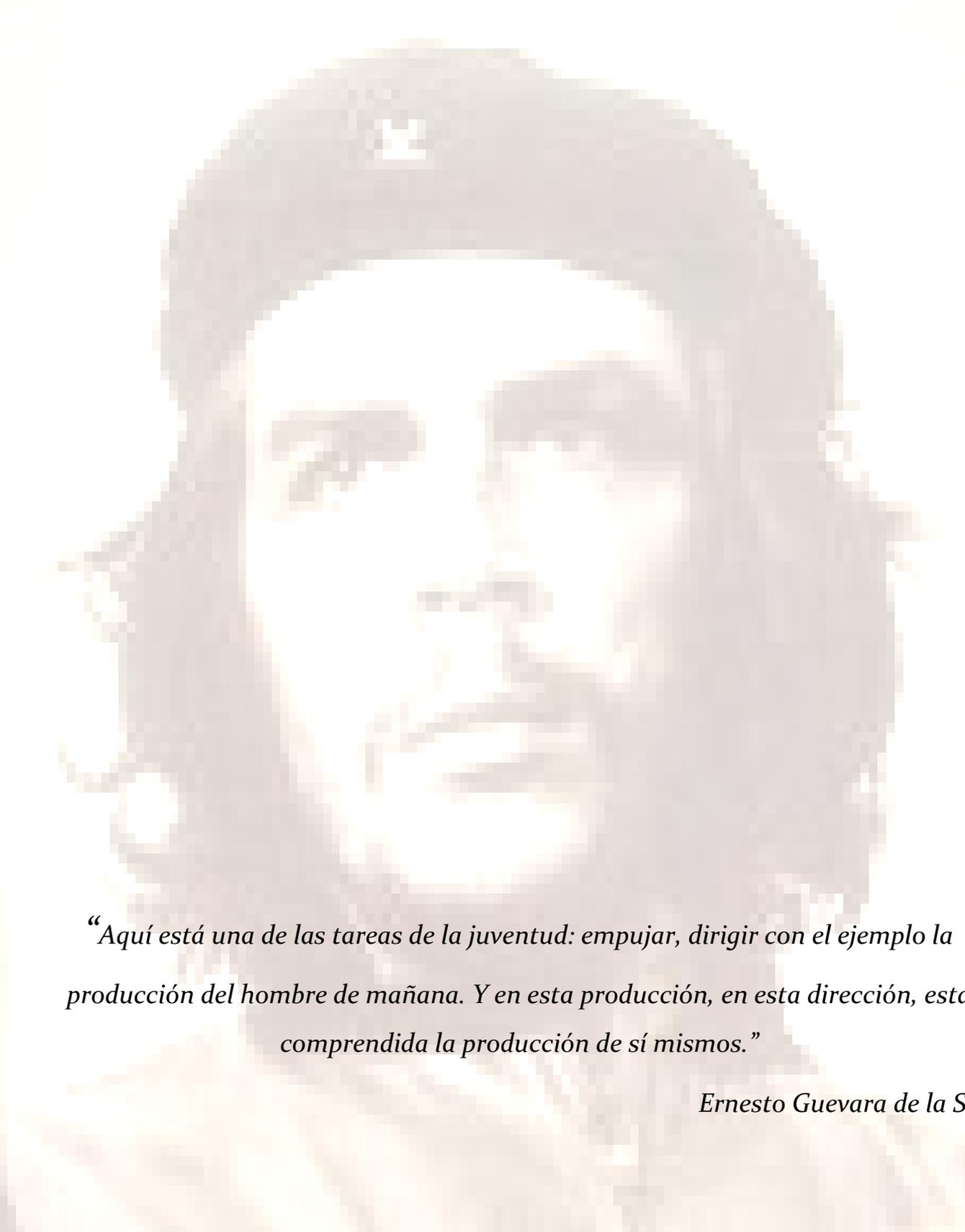
Autores: Lisett Morejón González.

Carmen Adriana Marrero Ramos.

Tutor: Ing. Norges Sánchez Tumbarell.

Ciudad de La Habana, Junio 2010

“Año del 52 Aniversario del Triunfo de la Revolución”



“Aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de sí mismos.”

Ernesto Guevara de la Serna

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Carmen Adriana Marrero Ramos

Lisett Morejón González

Firma del Autor

Firma del Autor

Ing. Norges Sánchez Tumbarell.

Firma del Tutor

DATOS DE CONTACTO

Ingeniero: Norges Sánchez Tumbarell.

Universidad de Ciencias Informáticas, Habana, Cuba.

E-mail: nsanchez@uci.cu

AGRADECIMIENTOS

De Lisett.

Agradecer a nuestro Comandante en Jefe y a la Revolución por darme la oportunidad de estudiar en esta Universidad de excelencia. A mi mamá por haber sido el motor impulsor para que yo ejerciera esta carrera, por su amor, su dedicación y paciencia, te quiero mucho mimiti. A pesar de que hace 4 años no veo a mi papá siempre ha estado conmigo y dándome ánimos para que saliera adelante y apoyándome en todo te quiero mucho y te extraño papito. A Camilo porque desde niña ha estado a mi lado dándome consejos y guiándome siempre por el buen camino, gracias por todo. A mis tres hermanos Lisbet, Adrian y Jorge Braulio por todo su amor y cariño pues siempre están ahí para mí dándome apoyo. A una de las personas más importantes en mi vida mi abuela Eugenia que aunque no está presente hoy, yo sé que está conmigo, pues gracias a ella hoy soy una ingeniera. A mi novio Javier por toda su paciencia, amor, dedicación te quiero mucho mi vida. A mis cuñados Paola y Yaisniel, a mis verdaderos amigos de universidad Yenis, Carmen y Yenier gracias por todo su cariño en estos 5 años ojalá y nos volvamos a encontrar después de terminar esta etapa de nuestras vidas. A mis eternas amigas que vienen conmigo desde el pre y algunas desde la primaria Lisi, Solainy, Anolandy, Yuslenys, Lizandra, Katy y Yoslainy gracias por todo su cariño. A Carmen por ser una gran compañera de tesis, por tener tanta paciencia, gracias. A mis dos tutores Norges Sánchez y Amari Jacson gracias por todo su apoyo para que la realización de esta tesis fuera posible. A mi familia en general mis tíos, primos por todo su amor durante mi carrera, a tía blanca por todo su apoyo y cariño durante toda mi vida. A mis segundos padres Fefa y Estebita que han sido un gran ejemplo para mí gracias por todo su amor los llevo siempre en mi corazón. A mis amistades de la universidad gracias por todos esos momentos que pasamos juntos nunca los voy a olvidar a Mara y Maelis gracias por todo. En fin a todas aquellas personas que en algún momento de sus vidas me dedicaron parte de su tiempo, gracias.

AGRADECIMIENTOS

De Carmen A.

Ante todo agradecer a mi mami pues siempre ha sido la luz que me ilumina, dándome todo el amor, confianza, apoyo y ánimo de seguir siempre adelante. Gracias mami por ser el regalo más bello que me ha dado la vida que de no ser por ti no hubiera podido lograr lo que he sido hasta hoy. Por esto y mucho más te quiero mucho.

A Tita y Escalona que los quiero como mis segundos padres ya que han estado ahí siempre que los necesito y sé que éste es también su sueño.

A mi familia en general en especial a mi abuela que aunque este lejos de mí siempre la llevo en mi corazón, a mis tías y en especial a mi tía chachi que aunque en este momento me hubiera gustado que estuviera junto a mí siempre me ha apoyado y brindado los mejores consejos del mundo, mis primas pues aunque no las veo mucho siempre las recuerdo y me han apoyado en el momento preciso.

A mis eternos amigos, de forma especial a Yenis, Lisett, Yenier y Javier que siempre han estado ahí para apoyarme y que me han hecho ver el camino más fácil cuando se ha tornado difícil, gracias de corazón y ojalá que el destino nunca nos separe. Nuevamente agradecer a Lisett por ser una excelente compañera de tesis pues juntas supimos lograr estos buenos resultados. A mis dos tutores, Norges Sánchez por habernos brindado su ayuda cuando más lo necesitábamos ayudándonos y apoyándonos en la realización de este trabajo. Amari que aunque no esté en estos momentos presente fue el primero que nos ayudó a dar los primeros pasos para el logro de nuestra tesis. A mis amistades de la universidad que en general son muchos y que aunque van cambiando y se renuevan al pasar del tiempo, cada uno de ellos deja una huella imborrable que es lo que hace que los recuerde siempre. Un beso para esas amistades que de una manera u otra siempre han estado presentes en momentos malos y buenos.

Y en general mil gracias a todas aquellas personas que de una forma u otra influyeron en algún momento en mi vida aportando siempre su granito de arena.

Con Amor, Carmen Adriana Marrero Ramos.

DEDICATORIA

De Lisett.

Dedico esta tesis a mis padres por todo el apoyo que me han dado pues sin ellos hoy no estuviese aquí, sobre todo a mi mamita y abuela que querían verme graduada, a mis hermanos por todo su cariño y comprensión y sobre todo a mi abuela Eugenia que sin ella nunca habría llegado hasta donde llegué hoy, por todo su amor y su dedicación hacia mí te quiero mucho.

De Carmen A.

Dedico este trabajo de manera especial:

A mi mami del alma por ser mi guía y apoyo incondicional durante toda mi vida, por luchar junto a mí hasta cumplir este sueño que también es suyo; gracias por estar siempre ahí, en todo momento junto a mí. A Tita y Escalona que son mis segundos padres y que siempre han querido verme graduada, ayudándome a ver todo lo bueno de esta vida. Y en general a todas aquellas personas que han estado cerca de mí dándome su apoyo incondicional.

RESUMEN

En la actualidad con el desarrollo de las tecnologías y el valor de la información que se genera constantemente, asociada a todas las esferas de la vida, se tienen inmensos volúmenes de datos. Si bien es cierto que la información que se maneja a diario permite la toma de decisiones, nos enfrentamos a los inconvenientes que se generan como consecuencia de la gran cantidad de datos que deben ser analizados. Como consecuencia de ello, en múltiples ocasiones es útil contar con herramientas que ayuden a minimizar el tiempo de análisis de este gran volumen de información y mejoren la eficiencia del trabajo con estos datos.

Existen las redes neuronales, las cuales son capaces de analizar altos volúmenes de información y brindarnos las mejores opciones, además incorporan nuevos criterios de análisis en la medida en que procesan los datos. Aunque las mismas han sido utilizadas fundamentalmente en tareas relacionadas con el reconocimiento de patrones y su clasificación, se propone su uso como una alternativa de solución eficiente para la toma de decisiones. En el presente se propone un Sistema de Análisis de la Información Académica que facilite la toma de decisiones, haciendo uso de las redes neuronales, donde puedan ser utilizados algoritmos existentes para obtener modelos explicables, particularmente en aquellos casos en los que la respuesta obtenida no es lo suficientemente concisa para nuestras necesidades.

Palabras Claves:

Redes neuronales, Arboles de Decisión, Arboles de Regresión, Trepan M5, Trepan Fuzzy.

Tabla de contenido

INTRODUCCIÓN.....	12
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.....	14
1.1 Introducción.....	14
1.2 Estado del Arte y Propuestas para el Desarrollo del Sistema.....	14
1.3 Metodología y Herramientas propuestas para la modelación del software.....	19
1.4 Herramientas.....	19
1.5 Lenguajes y Tecnologías.....	27
1.6 Conclusiones.....	30
CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA, ANÁLISIS Y DISEÑO.....	31
2.1 Introducción.....	31
2.2 Problema y situación problemática.....	31
2.3 Información que se maneja.....	31
2.4 Requisitos funcionales y no funcionales.....	32
2.4.1 Requisitos funcionales.....	32
2.4.2 Requisitos no funcionales.....	32
2.5 Propuesta del sistema.....	33
2.6 Modelo del Negocio.....	33
2.6.1 Actores y trabajadores que intervienen en el negocio.....	33
2.6.2 Diagrama de casos de uso del negocio.....	34
2.6.3 Descripción de los casos de uso del negocio.....	34
2.7 Especificación de los requisitos.....	35
2.8 Modelo del Sistema.....	35
2.8.1 Actores del sistema.....	35

2.8.2	Diagrama de caso de uso del Sistema.....	36
2.8.3	Descripción de los casos de uso del Sistema.	36
2.9	Análisis.	41
2.9.1	Modelo de Análisis.....	41
2.9.2	Diagrama de Clases del Análisis.	42
2.10	Modelo de diseño.	43
2.11	Patrones.	43
2.12	Diagramas de Clases del Diseño.....	43
2.13	Diagrama de Interacción.....	45
2.13.1	Diagrama de Secuencia.....	45
2.14	Diagrama de Despliegue.	48
2.15	Diagrama de Clases Persistentes.....	48
2.15.1	Descripción de las clases persistentes	49
2.16	Diseño de la Base de Datos.....	49
2.17	Prototipo de Interfaz de Usuario.....	50
2.18	Conclusiones.	50
CAPITULO 3: MÉTRICAS Y ESTUDIO DE FACTIBILIDAD.....		51
3.1	Métricas para la evaluación del diseño.	51
	Cantidad de procedimientos por clases	52
	Responsabilidad y Complejidad de las clases.....	53
3.2	Estudio de Factibilidad.....	58
3.2.1	Factibilidad Técnica	58
3.2.2	Factibilidad Económica	59
3.2.3	Factibilidad Operativa	60
3.3	Conclusiones.	61

Tabla de contenidos

Conclusiones	62
Recomendaciones	63
Bibliografía.....	64
Anexos.....	65
Anexo 1: Descripción de las clases persistentes.	65
Glosario de Términos.....	71

INTRODUCCIÓN

En el mundo actual debido al gran avance de las tecnologías son generados gran cantidad de volúmenes de datos. Si bien es cierto que la información que se maneja a diario permite la toma de decisiones, esto a su vez enfrenta diversos inconvenientes que se deben a la inmensa cantidad de datos que son analizados para obtener conclusiones específicas. Debido a esto sería eficaz contar con herramientas que nos faciliten lo antes planteado.

Las redes neuronales son capaces de analizar estos altos volúmenes de información brindando así las mejores opciones ya que pueden predecir con gran precisión.

En la actualidad existe diversa cantidad de software que hace uso de las redes neuronales. Entre estos se encuentran Software de Redes Neuronales para predicciones, los cuales fueron diseñados para ayudar a gerentes e ingenieros en la toma de decisiones para hallar predicciones y problemas de estimación, así como otros para el diseño de las redes neuronales artificiales el cual algunos son gratuitos pero otros son comerciales.

De lo anterior y como consecuencia de que en el país no contamos con ninguna herramienta que utilice redes neuronales para fines académicos, surge la necesidad de analizar grandes volúmenes de datos que contribuyan a la toma efectiva de decisiones y que a la vez sea capaz de aprender para futuros análisis.

Como base para resolver la necesidad planteada se propone un sistema que posibilite el análisis de grandes volúmenes de datos de manera eficiente y en el menor tiempo posible.

Para el logro exitoso de esta propuesta se han trazado los siguientes objetivos específicos:

- Proponer el Análisis y Diseño de un sistema que posibilite el análisis de grandes volúmenes de información, incluyendo los algoritmos más utilizados para dicho fin.

Para cumplir el objetivo anteriormente planteado se realizaron las siguientes **tareas investigativas**:

- Estudiar los diferentes clasificadores (árboles de decisión, de regresión, las redes neuronales y los modelos genéticos) y los criterios de evaluación de los mismos.
- Estudiar las arquitecturas y métodos de aprendizaje de las redes neuronales.(Tipos y Estructura)
- Extracción de reglas de una red neuronal. Métodos.

Para llevar a cabo las tareas planteadas se tiene como objeto de estudio Las redes neuronales, lo cual influyó sobre el análisis de grandes volúmenes de información.

Con esta investigación se obtuvieron los siguientes resultados:

- Obtención de Análisis y Diseño de un sistema que posibilita el análisis de grandes volúmenes de información, haciendo uso de las redes neuronales.
- Inclusión en la propuesta de solución del algoritmo TREPAN FUZZY¹ para extraer árboles difusos a partir de una red neuronal entrenada. Con este nuevo método es posible obtener árboles concisos y con un alto grado de flexibilidad a la red en cuanto a la colección de datos, para los que el árbol de decisión no es lo suficientemente conciso.
- Inclusión en la propuesta del algoritmo TREPAN M5² para extraer árboles modelos a partir de una red neuronal entrenada. Con este nuevo método es posible comprender el método de clasificación de una red neuronal entrenada para un rango no cubierto por TREPAN³.

Para mejor comprensión del documento, se estructuró en tres capítulos que se describen a continuación:

Capítulo 1- Fundamentación Teórica: Se realiza un estudio del estado del arte y de los antecedentes de las redes neuronales y minería de datos. Se describen las tendencias técnicas, las herramientas, tecnologías, metodologías y software usados en la actualidad escogida para la solución del problema.

Capítulo 2- Características del Sistema, Análisis y Diseño: Se expone una definición del objeto de estudio del problema, se plantean los objetivos estratégicos de la organización y procesos de negocio que los soportan. Se realiza una descripción de los procesos que serán objeto de automatización así como una descripción general de la propuesta de sistema, como debe funcionar. Definición del Modelo de negocio. Especificación de los requisitos del sistema y se define el Modelo de análisis. Comprende además las definiciones del Diseño realizándose los diferentes diagramas del mismo, así como los diagramas de secuencia y la descripción de las diferentes clases utilizadas.

Capítulo 3- Métricas y Estudio de Factibilidad: Se aplican métricas para la evaluación del diseño para verificar de una manera u otra la calidad y objetividad de los modelos de diseño. Comprende también un estudio de factibilidad para determinar la infraestructura tecnológica y la capacidad que implica la implantación del sistema.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción

A continuación se abordan los principales conceptos a tratar relacionados con las redes neuronales⁴ y se investiga la existencia de sistemas o herramientas empleadas en los procesos relacionados con la misma. Se estudian las tecnologías, metodologías y software con los cuales se realizará la propuesta de solución al problema antes planteado.

1.2 Estado del Arte y Propuestas para el Desarrollo del Sistema

Las redes neuronales surgieron del movimiento conexionista, que nació junto con la Inteligencia Artificial simbólica o tradicional. Esto fue hacia los años 50, con algunos de los primeros ordenadores de la época y las posibilidades que ofrecían. La IA simbólica se basa en que todo conocimiento se puede representar mediante combinaciones de símbolos, derivadas de otras combinaciones que representan verdades incuestionables o axiomas. Así pues, la IA⁵ tradicional asume que el conocimiento es independiente de la estructura que maneje los símbolos, siempre y cuando la 'máquina' realice algunas operaciones básicas entre ellos. En contraposición, los 'conexionistas' intentan representar el conocimiento desde el estrato más básico de la inteligencia: el estrato físico. Creen que el secreto para el aprendizaje y el conocimiento se halla directamente relacionado con la estructura del cerebro: concretamente con las neuronas y la interconexión entre ellas. (1)

Dentro de los objetivos que tienen es la de conseguir diseñar y construir máquinas capaces de realizar procesos con cierta inteligencia ha sido uno de los principales objetivos de los científicos a lo largo de la historia. De los intentos realizados en este sentido se han llegado a definir las líneas fundamentales para la obtención de máquinas inteligentes: En un principio los esfuerzos estuvieron dirigidos a la obtención de autómatas, en el sentido de máquinas que realizaran, con más o menos éxito, alguna función típica de los seres humanos. Hoy en día se continúa estudiando en ésta misma línea, con resultados sorprendentes, existen maneras de realizar procesos similares a los inteligentes y que podemos encuadrar dentro de la llamada IA. (2)

Esta tecnología puede ser desarrollada tanto en software como en hardware y con ella se pueden construir sistemas capaces de aprender, de adaptarse a condiciones variantes, o inclusive si se dispone de una colección suficiente grande de datos, predecir el estado futuro de algunos modelos. Las Redes

Neuronales (Neural Networks) son utilizadas para la predicción, la minería de datos (data mining), el reconocimiento de patrones y los sistemas de control adaptativo. (3)

Las redes neuronales son modelos simples de interconexiones neuronales. Una neurona artificial simula el proceso de un neurona en el cerebro humano. La misma son colecciones de nodos conectados, con entradas, salidas y procesamiento en cada nodo en donde la cual aprende desde un conjunto de entrenamiento desde el que generaliza patrones para clasificación y predicción.(3)

La utilización de la minería de datos no es nueva ya que desde los años sesenta estadísticos manejaban esta idea identificándola con otros términos como data fishing, data archaeology o como se llama actualmente data mining. Ya a principios de los años ochenta figuras como Rakesh Agrawal, Gio Wiederhold, Robert Blum y Gregory Piatetsky-Shapiro, entre otros, empezaron a consolidar estos términos. La misma consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información sería previamente desconocida resultando útil para algún proceso, más detalladamente la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos, también engloba todo un conjunto de técnicas encaminadas a la extracción de conocimiento procesable, implícito en las base de datos.(3)

La minería de datos conjuntamente con las redes neuronales es utilizada desde su inicio para la extracción de conocimiento a partir de los datos almacenados, la misma tiene aplicaciones en áreas financieras como en bancos, análisis de mercado, distribución, comercio, tratamientos médicos, predicción de respuestas de clientes y la identificación automatizada de objetos de interés en grandes bases de datos, también en el área educativa como son Pre-universitarios y universidades; es decir en todo aquello que requiera gran cantidad de almacenamiento de información. (2)

En la actualidad debido al gran avance tecnológico y al aumento de los volúmenes de información, referentes a múltiples esferas sociales, como es en el área educativa, se hace imprescindible el almacenamiento, la gestión oportuna y eficiente de los datos que se manejan, minimizando así los tiempos de análisis y las pérdidas que de ello pudieran generarse.

Actualmente existen empresas, corporaciones y universidades que se dedican al desarrollo de aplicaciones o herramientas para dichos fines, entre las que se destacan:

Apara es una compañía española que nace en el 2002 como spin off del Grupo Calidad y Tecnología. La misma es pionera y líder en España en el desarrollo de productos para la Gestión de la Decisión. Los productos de Apara, Dlife y DVelox, son capaces de transformar datos en información precisa que ayuden

en el proceso global de toma de decisiones. Estos satisfacen las necesidades de clientes de los sectores de Telecomunicaciones, Banca, Seguros, Salud y Administración Pública. (3)

Dlife / Apara es una plataforma bio-informática para la toma de decisiones clínicas. Proporciona un soporte computacional a la toma de decisiones médicas en los procesos de diagnóstico, tratamiento y seguimiento de la evolución de los pacientes que permite a los profesionales clínicos incrementar su precisión y la eficiencia de la prestación sanitaria en una media del 20%. (5)

En la universidad de Ingeniería de Chile existen varios Sistemas de Información Docente como lo es el de seguimiento curricular el cual los estudiantes pueden revisar cuantos cursos han realizado y viendo así si cumplen o no con lo solicitado. Este software facilita la administración de los egresos y es de gran ayuda para los alumnos en su planificación. Posteriormente se puso en práctica un sitio web (www.ing.puc.cl/dirdoc), en el que el usuario puede encontrar información actualizada y de manera rápida y amigable. (4) En la construcción de este nuevo sitio se plantearon dos objetivos fundamentales en primer lugar un buen diseño y organización pensando en el usuario y por último una estructura interna moderna que facilita su administración.

El objetivo es que los centro educativos, a partir de sus datos, puedan profundizar y predecir un futuro acerca del estudiante de acuerdo a su trayectoria académica para poder resolver mejor los problemas del mismo, como hacia donde irá el estudiante en un futuro y una vez que se encuentre cursando el último año como sería en su nuevo centro. (7)

Hoy en día, al igual que en el resto de ámbitos de la sociedad, las tecnologías de la información y las comunicaciones han irrumpido en el ámbito educativo. Así, es posible capturar y recopilar de forma muy sencilla y con coste reducido de todo tipo de información: datos administrativos de matriculaciones en colegios, institutos o universidades, expedientes académicos informatizados de los estudiantes, registro de actividad en portales educativos. (6)

La tecnología permite un nivel de colaboración aún más cercano al nivel de conocimiento. En el conjunto de herramientas conocido como Sopas de Conocimiento y el Gigante, desarrollado por la University of West Florida en colaboración con IBM. (8)

Latinoamérica como parte del Proyecto Quórum, los estudiantes colaboran compartiendo proposiciones de conocimiento durante la construcción de sus mapas conceptuales individuales, permitiendo además la crítica de las proposiciones por parte de otros estudiantes. Estas herramientas fueron presentadas en el VIII Congreso Internacional sobre Tecnología y Educación a Distancia en noviembre de 1997. (7)

Estas son algunas de las herramientas más importantes en el mundo de la minería de datos, ofreciendo facilidades para el desarrollo económico y educativo del país, sin dejar de destacar que son altamente costosas y producidas por empresas privadas, lo que trae consigo varios inconvenientes, entre los que se pueden mencionar: el código de las aplicaciones no es visible a las empresas o usuarios que adquieren estos sistemas, por lo que no existe garantía alguna de que la información que se maneja se encuentre totalmente segura; además no deben olvidarse las grandes sumas de dinero que hay que destinar a la compra de licencias y al mantenimiento de las aplicaciones.

Por todo lo antes planteado se llega a la conclusión de la necesidad de desarrollar sistemas propios que faciliten el manejo de grandes volúmenes de información de forma rápida y eficiente, que contribuyan a la toma efectiva de decisiones y que ayuden a obtener predicciones futuras.

Para ello se hace necesario el uso del software libre ya que es una manera segura de mantener nuestra información, pues este permite ver todo el código de la aplicación. El mismo es un movimiento tecnológico que ha revolucionado la sociedad presentando características especiales que han permitido la experimentación de nuevas formas de desarrollo y mantenimiento de programas, nuevos modelos económicos, educativos y nuevas normas legales. Es un asunto de libertad, no de precio. Para entender el concepto, se debe pensar en “libre” como en “libertad de expresión”.

El software libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, significa que los usuarios de programas tienen las cuatro libertades esenciales.

- La libertad de ejecutar el programa, para cualquier propósito (libertad 0).
- La libertad de estudiar cómo trabaja el programa, y cambiarlo para que haga lo que usted quiera (libertad 1). El acceso al código fuente es una condición necesaria para ello.
- La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).
- La libertad de distribuir copias de sus versiones modificadas a terceros (la 3ª libertad). Si lo hace, puede dar a toda la comunidad una oportunidad de beneficiarse de sus cambios. El acceso al código fuente es una condición necesaria para ello.

Un programa es software libre si los usuarios tienen todas esas libertades. Entonces, debería ser libre de redistribuir copias, tanto con o sin modificaciones, ya sea gratis o cobrando una tarifa por distribución, a cualquier persona en cualquier parte. El ser libre de hacer estas cosas significa, entre otras cosas, que no tiene que pedir o pagar el permiso, todo esto no significa que no sea comercial porque un programa libre

debe estar disponible para el uso comercial, la programación comercial y la distribución comercial. Esto da la medida de la viabilidad económica de este sistema libre por lo que evidentemente es la alternativa para los países subdesarrollados.

Principales Ventajas:

- **Bajo costo de adquisición y libre uso:** El usuario que adquiere software libre lo hace sin ninguna erogación monetaria o a muy bajo costo y ofrece un conjunto de recursos muy amplios. Cualquier persona con una computadora y una conexión a Internet puede utilizar un software libre. Para la mayoría de usuarios individuales el software libre es una opción atractiva por las libertades que garantiza sin necesidad de verse agobiados por el precio.
- **Innovación tecnológica:** El software libre, tiene como objetivo principal compartir la información, trabajando de manera cooperativa. Este es principalmente el modelo sobre el que la humanidad ha innovado y avanzado. Por lo tanto, los usuarios tienen un destacado papel al influir decisivamente en la dirección hacia donde evolucionan los programas: votando los errores que quieren que sean corregidos, proponiendo nueva funcionalidad al programa, o contribuyendo ellos mismos en el desarrollo del software.
- **Privacidad y Seguridad:** El Software Libre, por su carácter abierto, dificulta la introducción de código malicioso, espía o de control remoto, debido a que el código lo revisan muchos usuarios y desarrolladores que pueden detectar posibles puertas traseras. En el mundo del software libre, cualquier programador puede realizar una auditoría para comprobar que no se ha introducido ningún código malicioso, y, a su vez, cualquier entidad puede añadir libremente encriptación adicional a la aplicación que utilice para proteger sus datos.
- **Adaptación del software:** El software libre permite personalizar, gracias al hecho de que disponemos del código fuente, los programas tanto como sea necesario hasta que cubran exactamente nuestra necesidad.

En Cuba se ha desencadenado un auge nacional del software libre, principalmente en los órganos legislativos del estado y gobierno para garantizar la seguridad y la integridad de la información confidencial. El país se encuentra enfrascado en realizar un crecimiento en la industria del software, que da sus primeros pasos y cuya única alternativa sostenible para su avance, es la migración al software libre dándole un impulso tremendo a esta corriente de desarrollo.

En la actualidad existen ejemplos como lo es el portal <http://www.softwarelibre.cu/> en el cual se realizan diversas encuestas y se otorga la información con respecto a los talleres de Linux en la Isla. También tenemos la Universidad de las Ciencias Informáticas (UCI)⁶ la cual positivamente ha dedicado gran porcentaje al desarrollo de las tecnologías libres, ascendiendo en gran escala pues hoy cuenta con una distribución de GNU/Linux orientada a Escritorio llamada Nova que responde a las necesidades de las instituciones cubanas. Se llega a la conclusión de que con la utilización de las tecnologías libres Cuba será respetada y logrará la soberanía e independencia tecnológica.

1.3 Metodología y Herramientas propuestas para la modelación del software.

Dentro de la metodología de desarrollo de software se utilizará Rational Unified Process (RUP)⁷ pues este permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Constituyendo un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas. (5)

Para el modelado de esta aplicación se propone UML⁸ ya que está ampliamente difundido y permite visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, lo que posibilita entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

Como herramienta para modelar los diagramas en cada unas de las fases del desarrollo se utilizará Visual Paradigm para UML es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, permitiendo la captura de requisitos, análisis, diseño e implementación. Es un producto que facilita a las organizaciones diagramas visuales y diseño, integrar y desplegar sus aplicaciones empresariales de misión crítica y de sus bases de datos subyacentes. La herramienta ayuda a su equipo de desarrollo de software para sobresalir empleando una rápida respuesta y con poca memoria, que le permite manejar grandes y complicadas estructuras de un proyecto en una forma muy eficiente.

1.4 Herramientas

1.4.1 Lenguaje de programación del lado del servidor.

Existe una multitud de lenguajes cada uno de ellos explota más a fondo ciertas características que lo hacen más o menos útiles para los desarrolladores de software en la actualidad.

La versatilidad de un lenguaje está íntimamente relacionada con su complejidad. Un lenguaje complicado en su aprendizaje permite en general el realizar un espectro de tareas más amplio y más profundamente. Es por ello que a la hora de elegir el lenguaje que queremos utilizar tenemos que saber claramente qué es lo que queremos hacer y si el lenguaje en cuestión nos lo permite o no.

En el dominio de la red, los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP⁹, PHP¹⁰, y JSP¹¹, entre otros.

PHP

El PHP (acrónimo de PHP: Hipertexto Pre-processor), es un lenguaje interpretado de alto nivel embebido en páginas HTML¹² y ejecutado en el servidor. Es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI¹³, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

Ventajas:

- Posee una comunidad muy grande de desarrolladores, existen miles de lugares donde se pueden encontrar: documentación, tutoriales, ejemplos de código, foros etc.
- PHP es fácil de aprender comparado con otros lenguajes de programación. El lenguaje es semejante a C y Java pues la sintaxis primaria está basada en Perl. Además si se conoce JavaScript o ActionScript se verá la semejanza entre estos lenguajes, por ejemplo en sus estructuras de control.
- El rendimiento de PHP es muy bueno y verdaderamente eficiente, utilizando un servidor modesto puedes atender millones de peticiones al día.
- El precio para utilizar PHP es cero. Es gratuito y se puede descargar desde www.php.net.
- Es Open Source esto quiere decir que se tiene acceso al código fuente.
- Este lenguaje fue diseñado para trabajar sobre la Web por ello trae un conjunto muy amplio de funciones para ser utilizadas en diferentes tareas.

- Esta disponible para la mayoría de sistemas operativos existentes. Desde Unix, Linux, Microsoft Windows, MAC, entre otros.
- La versión 5 de PHP está diseñada para soporte de características de programación orientada a objetos. Características como herencia, métodos y atributos públicos o privados, clases y métodos abstractos, constructores, interfaces y destructores.
- Tiene soporte para conectarse a una gran variedad de base de datos como: MySQL, PostgreSQL, MySQL, Oracle, DBM, FilePro, HyperWave, Informix, InterBase, Sybase entre otras.

Desventajas:

- El manejo de errores no es tan sofisticado como en otros lenguajes.
- PHP consume más recursos y por tanto es ligeramente lento, llamando y ejecutando una función que ejecutando código que encuentra en línea, embebido script.

Por lo tanto después de haber hecho un estudio de los diferentes lenguajes de programación del lado del servidor podemos concluir que se propone **PHP**.

1.4.2 Gestor de Bases de Datos.

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- Definición de la información.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.

Algunos de los gestores de Bases de Datos que más se utilizan en el mundo y que es de software libre es PostgreSQL¹⁴.

PostgreSQL:

Es un sistema de gestión de bases de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. El desarrollo de PostgreSQL no es manejado por una sola empresa sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Ventajas:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs¹⁵ y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.
- Implementa el uso de rollback's¹⁶, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel.
- Es un producto Open Source¹⁷, sin costos de licencia.
- PostgreSQL tiene más de 15 años de desarrollo activo y se ha ganado la reputación de ser confiable y mantener la integridad de los datos.

Desventajas:

- Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- La velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, cosa que resulta loable.

Frameworks a utilizar.

El concepto de Frameworks¹⁸ no solo se emplea en aplicaciones Web sino que también está en muchos ámbitos del desarrollo de sistemas de software, podemos encontrar Frameworks para varios tipos de

aplicaciones como lo son el desarrollo de los juegos, la medicina así como otros ámbitos que se nos puedan ocurrir. Un Frameworks se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las piezas necesarias para construir una aplicación completa. En general con el término Frameworks nos referimos a una estructura de software compuesta de varios componentes personalizables e intercambiables para desarrollar una aplicación.

Los objetivos principales que persigue un Framework son:

- Acelerar el proceso de desarrollo.
- Reutilizar código ya existente.
- Promover buenas prácticas de desarrollo como el uso de patrones.

Algunos de los Frameworks más utilizados en la actualidad son para Java script ExtJS¹⁹, para Symfony²⁰, Doctrine, phpZend entre otros.

Symfony:

Symfony es el mejor Frameworks para crear aplicaciones PHP y la forma más sencilla de aumentar la productividad y calidad de tu trabajo. Symfony ha sido probado con éxito en algunos de los sitios web más grandes del mundo. Symfony aumenta exponencialmente tu productividad y te ayuda a mejorar la calidad de tus aplicaciones web aplicando todas las buenas prácticas y patrones de diseño que se han definido para la web.

Symfony es además el Frameworks más documentado del mundo, ya que cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales. También es Fácil de instalar y configurar en sistemas Windows, Mac y Linux, Funciona con todas las bases de datos comunes (MySQL²¹, PostgreSQL, SQLite²², Oracle²³, MS SQL Server²⁴), es Compatible solamente con PHP 5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP, Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización, Flexible hasta cualquier límite y extensible mediante un completo mecanismo de Plugins²⁵ y Traducido a más de 40 idiomas y fácilmente traducible a cualquier otro idioma.

Ventajas:

- Independiente del sistema gestor de bases de datos.

- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo
- Código fácil de leer.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros
- La capa de internacionalización que incluye Symfony permite la traducción de los datos y de la interfaz, así como la adaptación local de los contenidos.
- Nos permite reutilizar código entre los proyectos.
- Separación de modelo, vista y controlador (implementación MVC2)
- Dispone de una amplia documentación.

Desventajas:

- Fuerte curva de aprendizaje, aunque la documentación disponible facilita enormemente esta tarea. Este es un factor a tener en cuenta ya que la productividad del equipo de desarrollo no será alta en los primeros meses.
- Todavía hay muchas aplicaciones y desarrolladores que programan en PHP4. Esto puede plantear un problema ya que para extraer toda la potencia del Frameworks, el programador debe tener un buen conocimiento del paradigma de programación orientado a objetos.
- Aumenta el peso de la aplicación, ya que la implementación del patrón MVC²⁶ introduce varias capas de objetos que debe procesar el intérprete. La parte que más afecta al rendimiento es Propel, el sistema ORM²⁷ utilizado por defecto en Symfony.

ExtJS:

ExtJS es una librería de Java Script ligera y de alto rendimiento, es compatible con la mayoría de los navegadores y nos permite crear páginas e interfaces web dinámicas, es útil para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX²⁸, DHTML²⁹ y DOM³⁰. Esta librería incluye:

- Componentes UI del alto performance y personalizables.

- Modelo de componentes extensibles.
- Un API³¹ fácil de usar.
- Licencias Open Source y comerciales.

Una de las ventajas de ExtJS es que permite la creación de aplicaciones complejas utilizando componentes predefinidos así como un manejador de layout similar gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox³², Internet Explorer (IE)³³, Safari³⁴, etc.).

Además la ventana flotante que provee ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo cual le da una ventaja tremenda frente a otros.

Ventajas:

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

Desventajas:

- En el caso específico de ExtJS existe un factor que hay que tener en cuenta y que nace del hecho de ser una librería Java script independiente de cualquier tecnología del lado servidor. No existe una forma fácil de realizar Binding³⁵ entre los componentes visuales con el respectivo modelo, lo cual genera que el programador tenga que escribir más código para validar y enlazar los formularios.
- Es difícil hacer que el servidor haga push de información desde el servidor web hacia el navegador, haciendo que el cliente tenga que constantemente hacer Pooling³⁶ para obtener datos actualizados del servidor.

- La falta de un diseñador gráfico limita la difusión de la aplicación al no proveer una forma fácil y rápida de desarrollar en él, la situación me recuerda mucho a los inicios de Java Swing³⁷ donde la falta de un diseñador gráfico limitaba su alcance.
- Un problema grave del layout manager es que si usas código personalizado dentro de un componente, como por ejemplo un iframe, entonces se hace complicado lograr que se redimensione junto con el layout pues cae fuera de su control, sin embargo hay formas de lograrlo mediante código.
- Su sistema de licenciamiento no contempla la licencia LGPL³⁸, o tu código es 100% GPL³⁹ o debes pagar por su licencia de desarrollo.

CodeIgniter:

Es un framework para desarrollo de aplicaciones en PHP, es Open Source, muy pequeño (por lo menos comparándolo con Zend⁴⁰) y posee un acceso a sus librerías bien estructurado. Es liviano y fácil de instalar ya que sólo hay que descomprimirlo y copiarlo en una carpeta del servidor. CodeIgniter⁴¹ es más rápido pero menos fácil ya que carece de algunas librerías que los otros frameworks tienen, pero aún así no deja de ser un buen framework además de que es totalmente extensible y altamente compatible con gran variedad de versiones y configuraciones de PHP.

Ventajas:

- No necesitas saber algún lenguaje de consola para pre-instalar componentes.
- La documentación es clara y simple de entender, ya que está en un inglés sencillo.
- No se necesita un máster en php para entender las utilidades de CodeIgniter.
- CodeIgniter usa el patrón de diseño Modelo-Vista-Controlador como paradigma de arquitectura de desarrollo.

1.5 Lenguajes y Tecnologías.

1.5.1 Lenguajes de programación y tecnologías.

XML

Extensible Markup Language (XML)⁴² es un metalenguaje extensible de etiquetas desarrollado por el Word Wide Web Consortium. Es una simplificación y adaptación del Lenguaje de Mercado Generalizado (SGML)⁴³ y permite definir la gramática de lenguajes específicos. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

Ventajas:

- Es extensible: después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.

- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs⁴⁴ y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

HTML:

Hypertext Markup Language (HTML) 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 Java script), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

Java Script:

Fue creado en la empresa Netscape Communications, específicamente para su uso en el desarrollo de sitios Web. Es un lenguaje interpretado ya que no necesita de ningún compilador, solamente de un navegador, es orientado a objetos. Java Script es sumamente utilizado en el mundo del desarrollo Web por ser muy versátil y potente, tanto para la realización de pequeñas tareas como para la gestión de complejas aplicaciones. Con su utilización se puede evitar un poco la sobrecarga del lado del servidor ya que funciones como la validación de formularios, que antes se hacían del lado del servidor, se pueden realizar con este lenguaje y de una manera rápida.

Ventajas:

- El lenguaje de scripting es seguro y fiable.
- El código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright. Esto, en nuestra opinión es una ventaja.
- El código Java Script se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; un script⁴⁵ ejecutado en el servidor, sin embargo, sometería a éste a dura prueba y los servidores de capacidades más limitadas podrían resentir de una continua solicitud por un mayor número de usuarios.

Desventajas:

- Los script tienen capacidades limitadas, por razones de seguridad.
- El código del script debe descargarse completamente antes de poderse ejecutar y ésta es la otra cara de la moneda, si los datos que un script utiliza son muchos, el tiempo que tardará en descargarse será muy largo, mientras que la interrogación de la misma base de datos en el servidor sería más rápida.

1.5.2 *Propuesta de los algoritmos a utilizar en la solución.*

Trepan M5: Construye árboles modelos, regresiones lineales⁴⁶ y árboles de regresión⁴⁷ utilizando una red neuronal (u otro clasificador a elección) como oráculo. El algoritmo M5 es una técnica de aprendizaje supervisado el cual utiliza la siguiente idea: dividir el espacio de datos en áreas (sub-espacios) y construir un modelo de regresión lineal en cada una de ellas. El modelo resultante puede verse como un modelo modular formado por modelos lineales especializados (modelos “locales”) es frecuentemente utilizada.

El algoritmo M5 se basa en el principio de la teoría de información, el test de participación es similar al de un árbol de decisión⁴⁸, pero en lugar de clases asigna funciones de regresiones lineales a las hojas del árbol. Los árboles modelos generalizan el concepto de los árboles de regresión, que tienen valores constantes en sus hojas. Los árboles modelos son eficientes y pueden manejar dominios de grandes dimensiones y cientos de atributos. La mayor ventaja de los árboles modelos sobre los arboles de regresión es que los árboles modelos son mucho más compactos que los de regresión y son más precisos. Además las funciones de regresión normalmente no involucran muchas variables. (1)

Trepan Fuzzy: Este algoritmo construye árboles difusos utilizando una red neuronal (u otro clasificador a elección) como oráculo⁴⁹. Este método de inducción se basa en la discretización “suave” de los atributos continuos a conjuntos difusos. En este algoritmo se obtienen clasificadores precisos y con alta fidelidad. De acuerdo a estos resultados los usuarios pueden tomar la decisión final o realizar una investigación más profunda. (1)

Regresiones Lineales: Una función de regresión lineal modela la salida de la clasificación como una combinación lineal de los atributos de entrada. Las regresiones lineales se utilizan para clasificar dataset con clases numéricas.

Árboles de decisión: Un árbol de decisión es un discriminador de clases que recursivamente particiona un conjunto de instancias hasta que cada partición consista íntegramente o predominantemente de ejemplos de una clase. Cada nodo interno de un árbol contiene de separación o criterio de partición el que consiste

de un test sobre uno o más atributos y determina el particionamiento de los datos. Los árboles de decisión representan reglas.

Árboles de Regresión: Tiene una estructura similar a la de un árbol de decisión, pero sus hojas son valores reales en lugar de identificadores de clases. Al igual que las regresiones lineales se utilizan para clasificar dataset con clases numéricas.

1.6 Conclusiones

Se hizo un estudio y valoración de las principales herramientas, metodologías, y lenguajes que se utilizan para el desarrollo de proyectos informáticos en la actualidad, así como de las principales herramientas existentes en el mundo que satisfagan la necesidad de nuestra proposición. Después de haber realizado todo el estudio del arte, se hizo una propuesta para llevar a cabo el análisis y diseño de un sistema que posibilite el análisis de grandes volúmenes de información de manera eficiente y en el menor tiempo posible.

CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA, ANÁLISIS Y DISEÑO.

2.1 Introducción

Se hace una propuesta para analizar grandes volúmenes de datos utilizando las redes neuronales, a partir del problema planteado. Se analiza la situación actual, se presenta el modelo de negocio describiéndose los diagramas de caso de uso del negocio, los diagramas de actividades para cada uno de estos, se demuestran los requerimientos funcionales como los no funcionales y se define cuales son los actores y los caso de uso que intervienen en el sistema así como el diseño que se relaciona con la creación de una solución para el problema analizado; un modelo es una simplificación utilizada para comprender mejor el problema (modelo de análisis) o la solución (modelo de diseño)”.

2.2 Problema y situación problemática.

Actualmente en la universidad de ciencias informáticas se maneja gran parte de información a través de un sistema automatizado de gestión académica – akademos. Este sistema cuenta con varios módulos que te permiten adquirir datos acerca del estudiante pero el mismo no favorece totalmente a la toma de decisiones para algunos directivos de una determinada facultad cuando se necesario saber antecedentes y avances de un estudiante de la universidad en un momento determinado, sacando predicciones futuras con la ayuda de las redes neuronales conjuntamente con algoritmos que son utilizados en la inteligencia artificial que ayudan a alcanzar objetivos concretos.

2.3 Información que se maneja.

Se gestiona toda la información que necesiten los distintos niveles de dirección de la Universidad, ya sean vicerrectores, secretaria general, decanos, secretarías docentes, etc. Cada curso escolar en la universidad se hace necesario conformar un resumen donde se explica el comportamiento de los estudiantes a partir de notas y trayectorias durante su estancia en el centro. A estos directivos se les hace engorroso el trabajo pues tiene que ser manualmente y exponen en el claustro mediante presentaciones realizadas por ellos mismos haciendo el proceso largo y trabajoso.

2.4 *Requisitos funcionales y no funcionales.*

2.4.1 *Requisitos funcionales.*

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir.

Los requisitos funcionales que se identificaron son los siguientes:

- R1.** Permitir al usuario entrar al sistema.
- R2.** Permitir a los usuarios obtener y consultar información.
- R3.** Permitir al administrador definir permisos y cuentas de usuarios.
- R4.** Permitir al usuario configurar la entrada de datos para el análisis.
- R5.** Permitir visualizar los resultados obtenidos.

2.4.2 *Requisitos no funcionales.*

Los requisitos no funcionales son cualidades que el producto debe tener. Debe pensarse en estas cualidades como las características del entorno que hacen al producto atractivo, usable, rápido o confiable.

El sistema debe cumplir ciertas características, tales como:

RNF1. Apariencia o interfaz externa.

La aplicación presentará un color que denote seriedad debido a su contenido además deberá ser atractivo a la vista del usuario, se seleccionará como color predominante el blanco.

RNF2. Usabilidad.

En la aplicación se garantizará un acceso fácil y rápido a los programadores. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y de un ambiente Web en sentido general.

RNF3. Rendimiento.

Garantizará tiempos de respuestas y velocidad de procesamiento de la información generalmente rápidos.

RNF4. Software.

La máquina servidor deberá disponer del servidor Web Apache versión 2.0. Además se debe constar con el sistema gestor de base de datos PostgreSQL. El lenguaje de programación debe ser PHP en su versión 4 o más avanzada

Las máquinas clientes podrán contar con sistema operativo Windows XP o superior o cualquier distribución de Linux⁵⁰. Las máquinas clientes para acceder al sistema deben hacerlo a través del navegador web Mozilla o Internet Explorer.

RNF5. Hardware.

La máquina servidor deberán tener como mínimo 40 GB de capacidad de disco duro, microprocesador superior a 1.00 GHz, 512 mgb mínimo de memoria RAM.

RNF6. Requisitos para la documentación de usuarios en línea y ayuda del sistema.

Se realizará un Manual de Usuario para ayudarlos a trabajar con el sistema.

2.5 Propuesta del sistema.

A partir de lo antes planteado se propone una nueva alternativa que maneje los datos de los alumnos y ayude a los directivos de la universidad a la toma de decisiones y al arribo de conclusiones de cómo el estudiantes puede comportarse en un futuro con la ayuda de algoritmos y con el uso de las redes neuronales que tienen como objetivo obtener predicciones a partir de información almacenada.

2.6 Modelo del Negocio.

“El modelo del negocio se realiza con el propósito de comprender las características y actividades que se llevan a cabo en el contexto a automatizar. En él, se modelan los clientes y trabajadores del negocio y su interacción con los procesos y objetos que lo componen. Está formado por el modelo de casos de uso del negocio y el modelo de objetos del negocio.” (COLLERA 2005)

2.6.1 Actores y trabajadores que intervienen en el negocio.

Un actor del negocio es cualquier individuo, grupo, organización o máquina que interactúa con el negocio.

Un trabajador del negocio se utiliza en el modelado de negocio como punto de partida para derivar un primer conjunto de actores y casos de uso del sistema de información en construcción.

Actores y Trabajadores del Negocio

Actores del negocio	Justificación
Usuario	Interviene en el proceso de la confección del resumen donde se expone el estado del estudiante durante un periodo determinado.

Tabla 1 Actores del negocio

Trabajadores del negocio	Justificación
Sistema Akademos	Almacena toda la información acerca de los estudiantes de la universidad.

Tabla 2 Trabajadores del negocio

2.6.2 Diagrama de casos de uso del negocio.

El modelo del negocio describe el negocio en términos de casos de usos del negocio, que corresponde a lo que generalmente se le llama procesos.

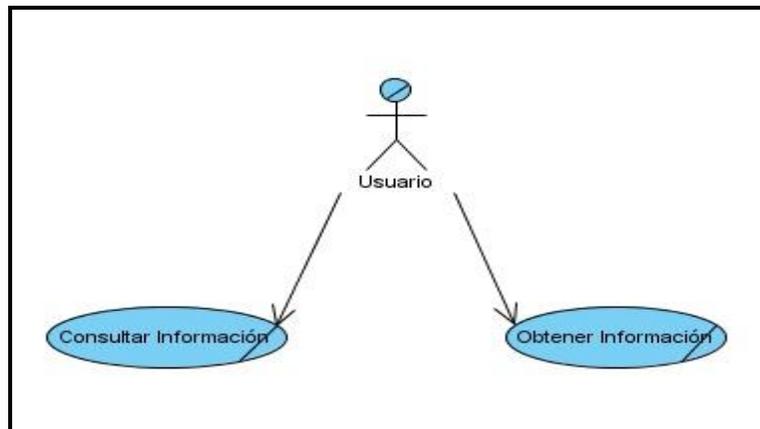


Fig. 1 Diagrama de casos de uso del negocio

2.6.3 Descripción de los casos de uso del negocio.

Un caso de uso del negocio describe el comportamiento de una sucesión de acciones que produce un resultado de valor para un actor particular del negocio.

Descripción de los casos de uso del negocio.

Nombre del Caso de Uso: Obtener Información	
Actores:	Usuario
Propósito:	Obtener información que necesita el actor.
Resumen:	El caso de uso se inicia cuando el usuario tiene permiso para obtener información.
Referencias:	R2

Tabla 3 Descripción del caso de uso Obtener Información

Nombre del Caso de Uso: Consultar Información	
Actores:	Usuario
Propósito:	Consultar Información que necesita el actor
Resumen:	El caso de uso se inicia cuando el actor consulta la información
Referencias:	R2

Tabla 4 Descripción del caso de uso Consultar Información

2.7 Especificación de los requisitos.

El objetivo de la captura de requisitos es guiar el desarrollo de software hacia el sistema correcto, definiendo objetivos generales concretos de manera tal que tanto el negocio como sus actores se beneficien. Se utilizaron dos técnicas para lograr este objetivo: la entrevista y los cuestionarios.

2.8 Modelo del Sistema.

2.8.1 Actores del sistema.

Los actores se corresponden en la mayoría de los casos con trabajadores y actores del negocio y representan los usuarios del sistema y otras aplicaciones que interactúan con él.

Actores del sistema

Actores del sistema	Justificación
Vicedecano de Formación	Interviene en el proceso de la confección del resumen donde se expone el estado del estudiante durante el curso escolar.
Administrador	Gestionar las cuentas de los usuarios del sistema.

Tabla 5 Actores del sistema

2.8.2 Diagrama de caso de uso del Sistema.

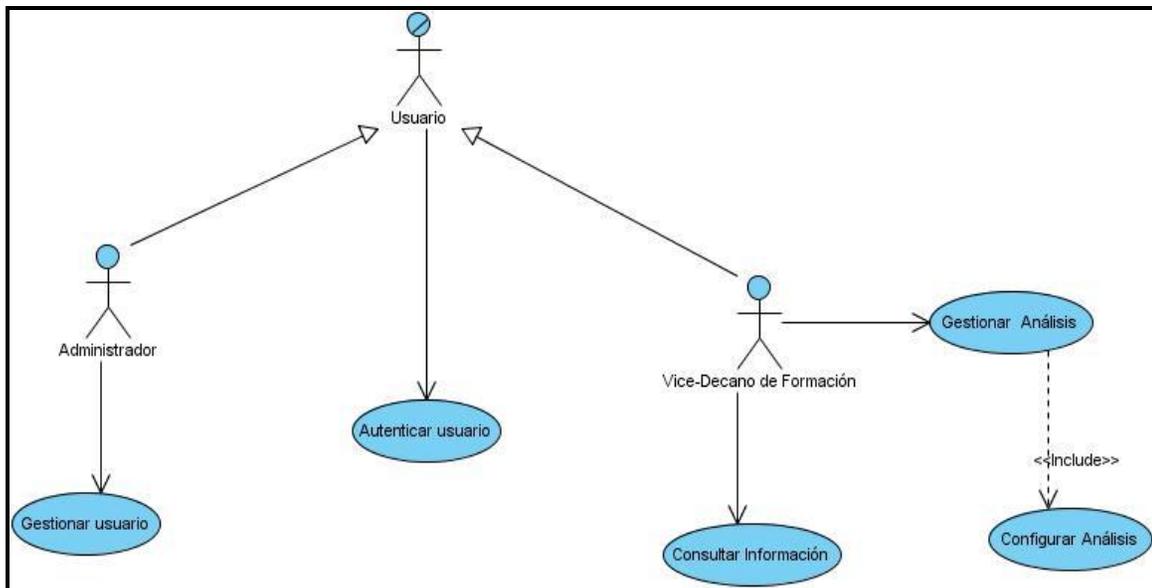


Fig. 2 Diagrama de casos de uso del sistema

2.8.3 Descripción de los casos de uso del Sistema.

Descripción del caso de uso Gestionar usuario

Nombre del Caso de Uso: Gestionar usuario	
Actores:	Administrador
Propósito:	Permitir al actor gestionar los usuarios que van a interactuar con el sistema.

Capítulo 2: Características del sistema, Análisis y Diseño

Resumen:	El caso de uso se inicia cuando el administrador necesita gestionar una cuenta de usuario (adicionar nueva cuenta, modificarla, eliminarla o buscarla. El caso de uso concluye cuando finalizan cualquiera de los escenarios de la gestión de la cuenta escogida por el actor.	
Referencias:	R1,R3	
Precondiciones:	El actor se ha autenticado correctamente en el sistema. El usuario debe tener los permisos necesarios para realizar la operación.	
Flujo Normal de Eventos		
Sección 1. Buscar usuario		
Acción del Actor	Respuesta del sistema	
1. Llena los criterios de búsqueda y le da a la opción Buscar.	2. Se muestra el listado según los criterios especificados, permitiendo eliminar cualquier usuario obtenido. 3. Termina el caso de uso.	
Flujo Alterno		
El usuario selecciona la opción Cancelar en cualquier momento y el sistema no muestra los datos de los usuarios y termina el caso de uso.		
Sección 2. Adicionar usuario		
Acción del Actor	Respuesta del sistema	
4. Escoge la opción adicionar una nueva cuenta de usuario.	5. Habilita el formulario para llenar los datos necesarios de la cuenta.	
6. Llena el formulario y presiona el botón Aceptar.	7. El sistema valida y guarda los nuevos datos, habilitándose la nueva cuenta.	
Flujo Alterno		
Línea 6.		
7.1 El sistema detecta errores en los datos adicionados mostrando un mensaje “Datos incorrectos”		

7.2 El administrador arregla los datos incorrectos y adiciona nuevamente.	
Sección 3. Eliminar usuario	
Acción del Actor	Respuesta del sistema
8. Selecciona en los elementos del resultado de la búsqueda la opción de eliminar una cuenta de un usuario en específico.	9. Muestra un mensaje “Desea eliminar al usuario” .
10. El usuario selecciona la opción Aceptar.	11. El sistema borra la cuenta de la base de datos y muestra un mensaje “Usuario eliminado satisfactoriamente” . 12. Termina el caso de uso.
Flujo Alterno	
<p>Línea 8.</p> <p>8.1 Si no hay una cuenta de usuario seleccionada el sistema muestra el mensaje “Seleccionar cuenta de usuario”, regresando a la línea 7 de la sección 3.</p> <p>Línea 8</p> <p>9.1 El usuario selecciona Cancelar y el sistema no elimina los datos. Termina el caso de uso.</p>	
Sección 4. Modificar usuario	
Acción del Actor	Respuesta del sistema
13. Selecciona en los elementos del resultado de la búsqueda la opción de modificar una cuenta de un usuario en específico.	14. Muestra un formulario con los datos del usuario a modificar.
15. El usuario modifica los datos que desea y presiona el botón Aceptar.	16. El sistema valida y guarda los datos. 17. Termina el caso de uso

Flujo Alterno	
Línea 13	
13.1 Si no hay cuenta de usuario seleccionada el sistema muestra un mensaje “ Seleccionar cuenta de usuario ”, regresando a la línea 11 de la sección 4.	
Línea 15	
15.1 El sistema detecta errores en los datos modificados mostrando un mensaje “ Datos incorrectos ”.	
15.2 El administrador arregla los datos incorrectos y solicita guardar los datos nuevamente, regresando a la línea 15 de la sección 4.	
15.3 El usuario selecciona la opción Cancelar en cualquier momento y el sistema no modifica ningún dato.	
Termina el caso de uso.	
Pos condiciones:	Todos los usuarios han sido gestionados(mostrados, adicionados, eliminados o modificados)
Prioridad:	Crítico

Tabla 6 Descripción del caso de uso Gestionar usuario

Descripción del caso de uso Autenticar usuario

Nombre del Caso de Uso: Autenticar usuario	
Actores:	Usuario
Propósito:	Autenticar usuario al sistema.
Resumen:	El actor introduce su usuario y contraseña. El sistema le permite la posibilidad de acceder. EL caso de uso termina cuando el usuario accede al sistema satisfactoriamente.
Referencias:	R1
Precondiciones:	El actor debe estar registrado en el sistema previamente.
Flujo Normal de Eventos	

Acción del Actor	Respuesta del sistema
1. Introduce el usuario y contraseña y presiona Aceptar.	2. El sistema valida los datos. 3. El sistema muestra interfaz principal según el rol del usuario. 4. Termina el caso de uso
Flujo Alternativo	
Línea 2	
2.1 El sistema detecta errores en los datos introducidos y los notifica al usuario solicitando que sean rectificadas.	
2.2 El actor rectifica nuevamente su usuario y contraseña y presiona Aceptar, regresando a la línea 2.	
2.3 El selecciona la opción Cancelar y termina el caso de uso.	
Pos condiciones:	El usuario se ha autenticado satisfactoriamente.
Prioridad:	Crítico

Tabla 7 Descripción del caso de uso Autenticar usuario

Descripción del caso de uso Gestionar Información

Nombre del Caso de Uso: Gestionar Información	
Actores:	Vice-Decano de Formación
Propósito:	Permitir gestionar toda la información necesaria para un posterior análisis.
Resumen:	El caso de uso se inicia cuando el vice-decano de formación gestiona toda la información necesaria para el análisis a partir de los datos obtenidos. El caso de uso concluye cuando finalizan cualquiera de los escenarios de la gestión de análisis escogidos por el actor.
Referencias:	R2, R4, R5
Precondiciones:	El usuario debe tener los permisos necesarios para manejar la

	información del sistema.
Flujo Normal de Eventos	
Sección 1. Consultar Datos	
Acción del Actor	Respuesta del sistema
1. El usuario consulta la información.	2. Muestra la Información. 3. Termina el caso de uso.
Sección 2. Gestionar Análisis	
Acción del Actor	Respuesta del sistema
4. Selecciona los datos necesarios, el rango de los datos y los campos de orden para el desarrollo del análisis.	5. Valida los datos. 6. Guarda todos los datos seleccionados. 7. Termina el caso de uso.
Flujo Alterno	
5.1 El sistema detecta errores en los datos almacenados mostrando un mensaje “ Datos incorrectos ”	
5.2 El Vice-Decano arregla los datos incorrectos y guarda nuevamente.	
Pos condiciones:	El sistema guardó satisfactoriamente toda la información obtenida.
Prioridad:	Crítico

Tabla 8 Descripción del caso de uso Gestionar Información

2.9 Análisis.

En el análisis se refinan y estructuran los requisitos obtenidos con anterioridad, se profundiza en el dominio de la aplicación lo que permite una mayor comprensión del problema para modelar la solución.

2.9.1 Modelo de Análisis.

El análisis consiste en obtener una visión del sistema, que se preocupa de ver que hace, solo se interesa por los requisitos funcionales.

2.9.2 Diagrama de Clases del Análisis.

En el Diagrama de clases del análisis se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas. A continuación se muestran algunos diagramas de clases del análisis.

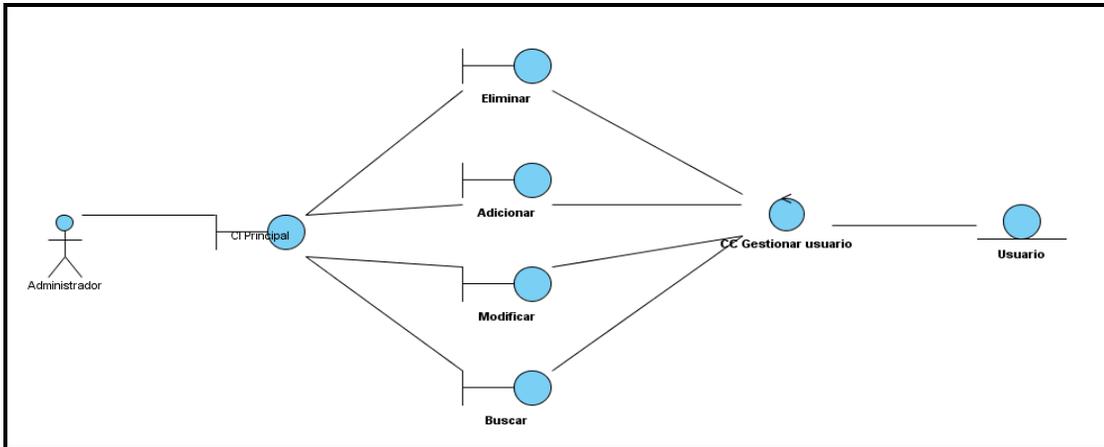


Fig. 3 Diagrama de clases Gestionar Usuario

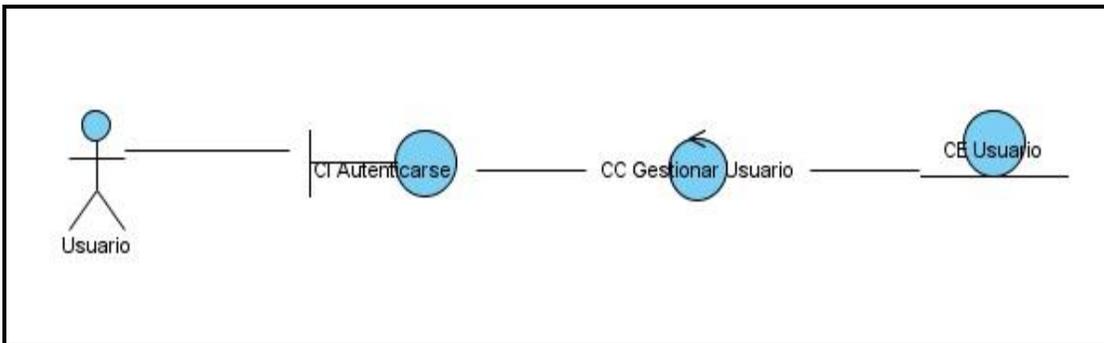


Fig. 4 Diagrama de clases Autenticar Usuario

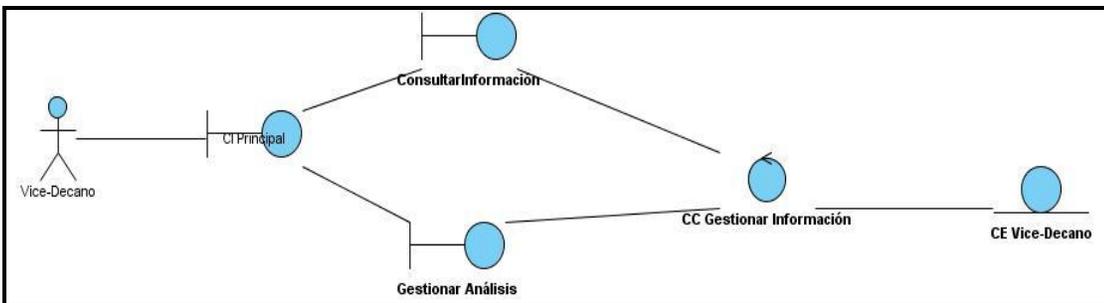


Fig. 5 Diagrama de clases Gestionar Información

2.10 Modelo de diseño.

En el diseño se modela el sistema y se encuentra su forma (incluida la arquitectura) para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Una entrada esencial en el diseño es el resultado del análisis, que proporciona una comprensión detallada de los requisitos.

2.11 Patrones.

Los patrones de diseño son soluciones a problemas ya conocidos que ayudan a un mejor rendimiento, desarrollo y mantenimiento del software donde se apliquen, por lo que se estudió la posibilidad de aplicarlos. Se propone utilizar el patrón MVC (modelo-vista-controlador), que implementa por defecto, pero aprovechándolo al máximo. Este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página; el modelo es el sistema de gestión de base de datos y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Vista.

Página servidora (SP Página), página cliente (CP Página) y hoja de estilo (Hoja de Estilo).

Controlador.

El “codebehind” (CB Página), el “codebehind” (CB Control Ajax), clase controladora (Control), Java script (Java script) y página servidora pura (Control Ajax).

2.12 Diagramas de Clases del Diseño

Los diagramas de clases del diseño son una representación más concreta y detallada que los diagramas de clases del análisis, aunque también representan la parte estática del sistema conteniendo las clases y sus relaciones.

A continuación se presentan los diagramas de clases del diseño de los casos de usos críticos del sistema propuesto:

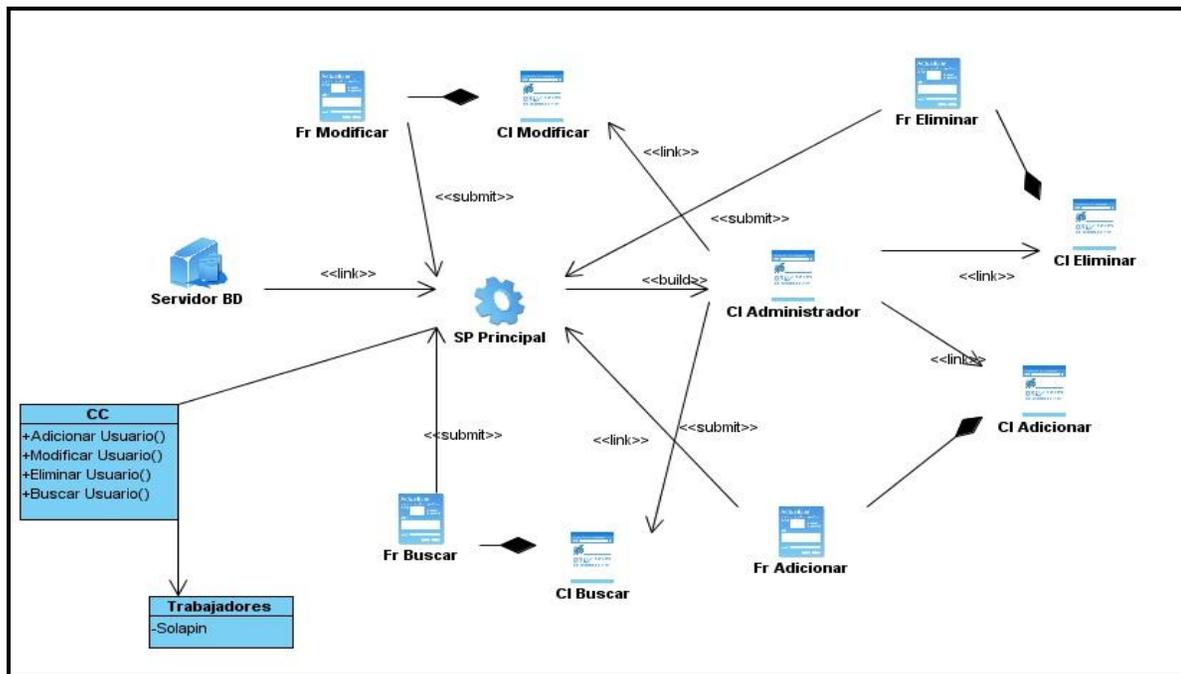


Fig. 6 Diagrama de clases del diseño: Caso de Uso Gestionar Usuario

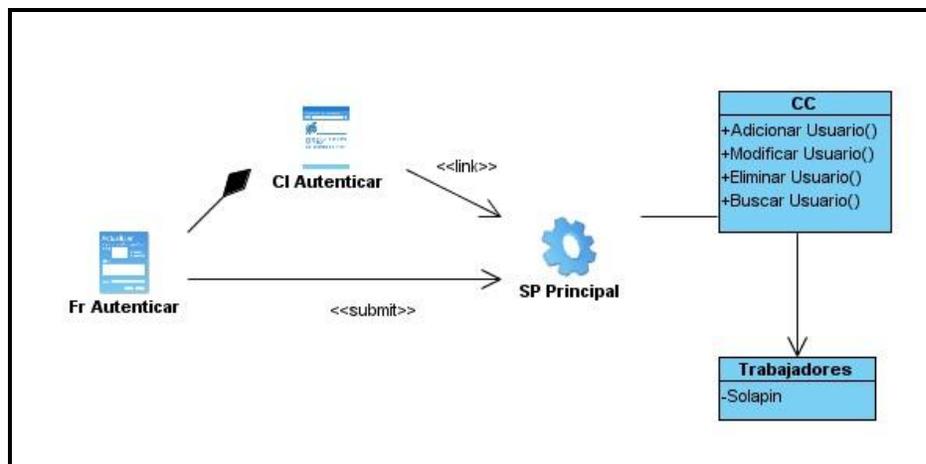


Fig. 7 Diagrama de Clases del Diseño: Caso de Uso Autenticar Usuario

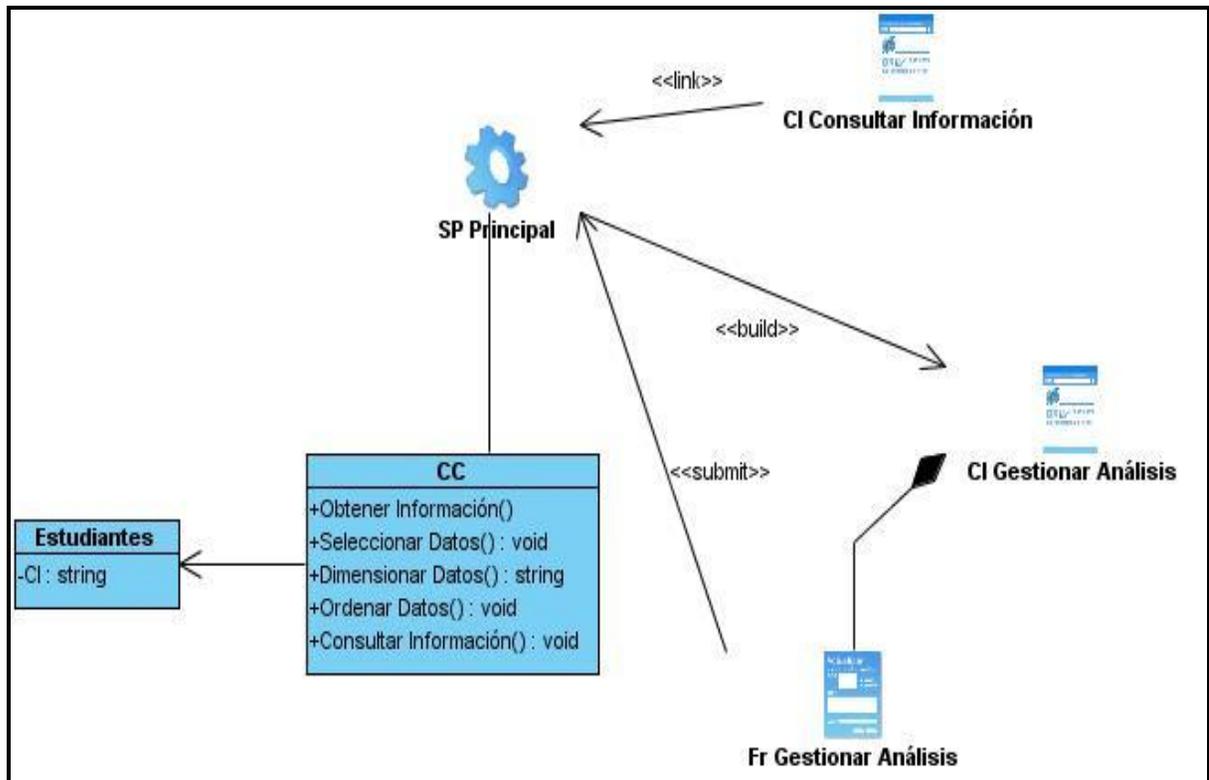


Fig. 8 Diagrama de Clases del Diseño: Caso de Uso Gestionar Información

2.13 Diagrama de Interacción.

Los diagramas de secuencia y los diagramas de colaboración (ambos llamados diagramas de interacción) se utilizan para modelar los aspectos dinámicos de los sistemas.

2.13.1 Diagrama de Secuencia.

Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes.

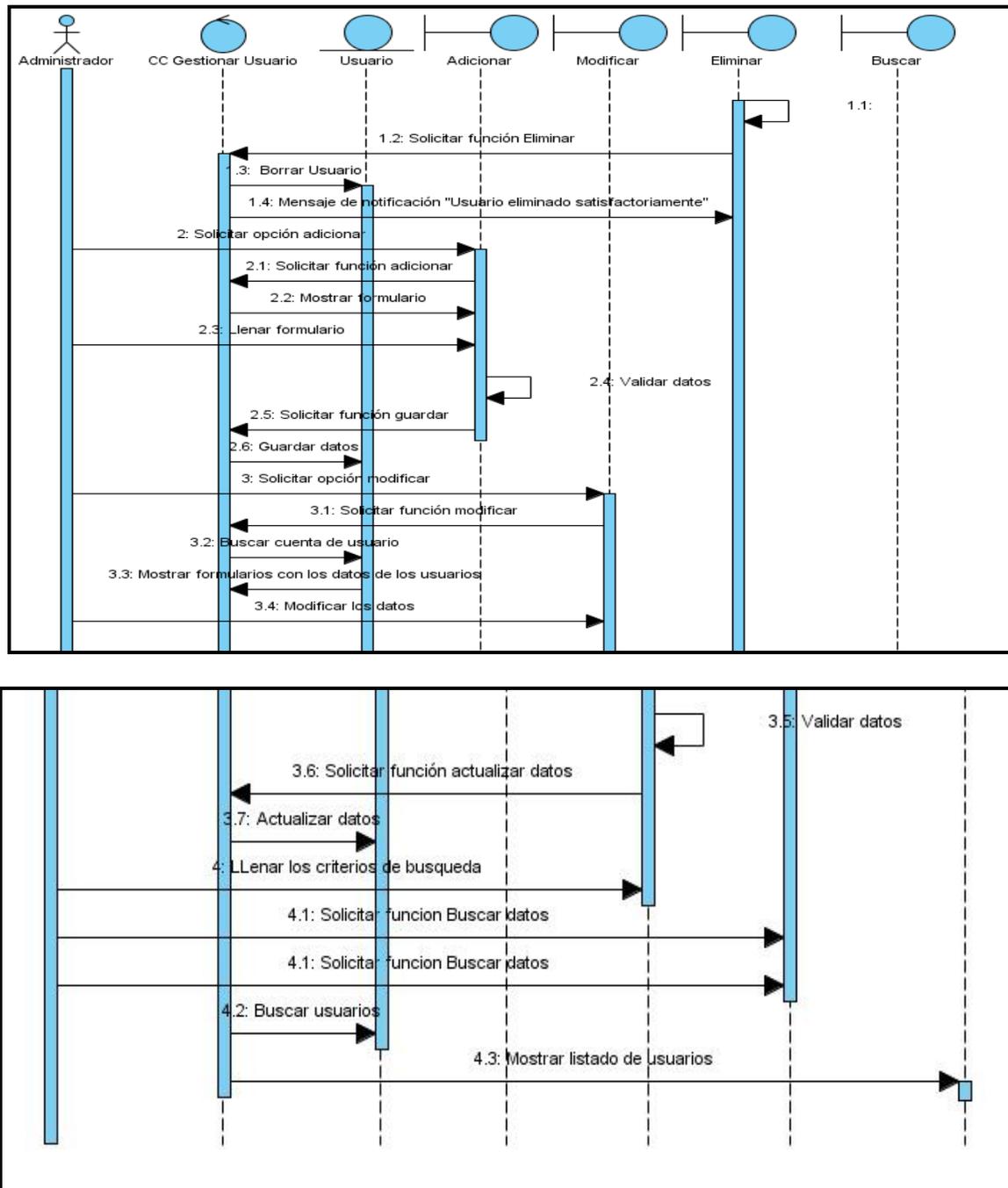


Fig. 9 Diagrama de Secuencia: Caso de Uso Gestionar Usuario

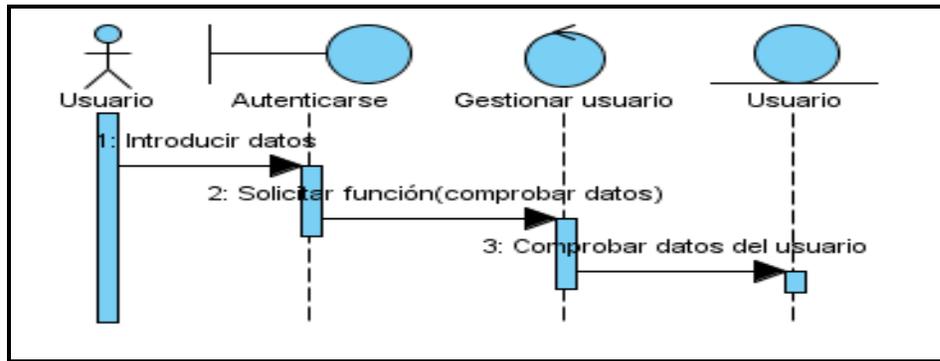


Fig. 10 Diagrama de Secuencia: Caso de Uso Autenticar Usuario.

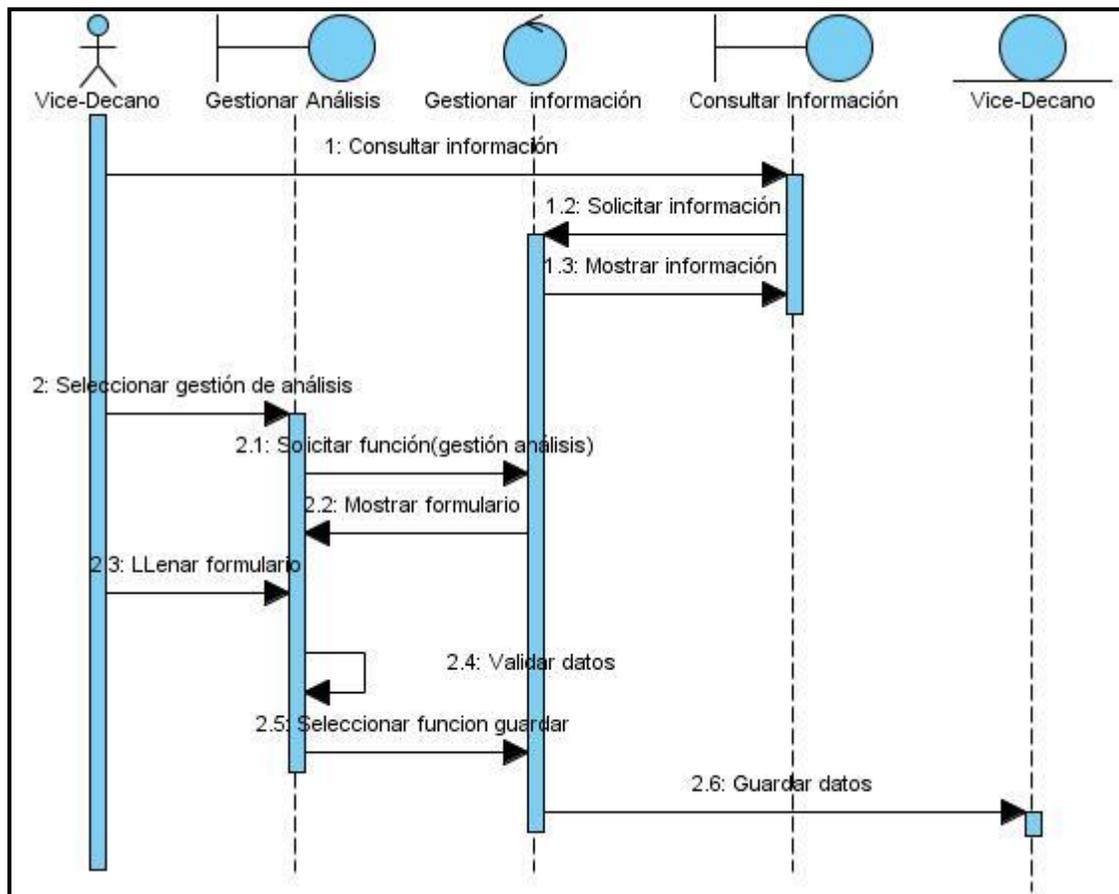


Fig. 11 : Diagrama de Secuencia: Caso de Uso Gestionar Información

2.14 Diagrama de Despliegue.

El diagrama de despliegue describe los componentes, dispositivos, hardware y software que se desplegará en el ambiente seleccionado. A continuación se presenta como el sistema propuesto queda distribuido:



Fig. 12 Diagrama de Despliegue

2.15 Diagrama de Clases Persistentes.

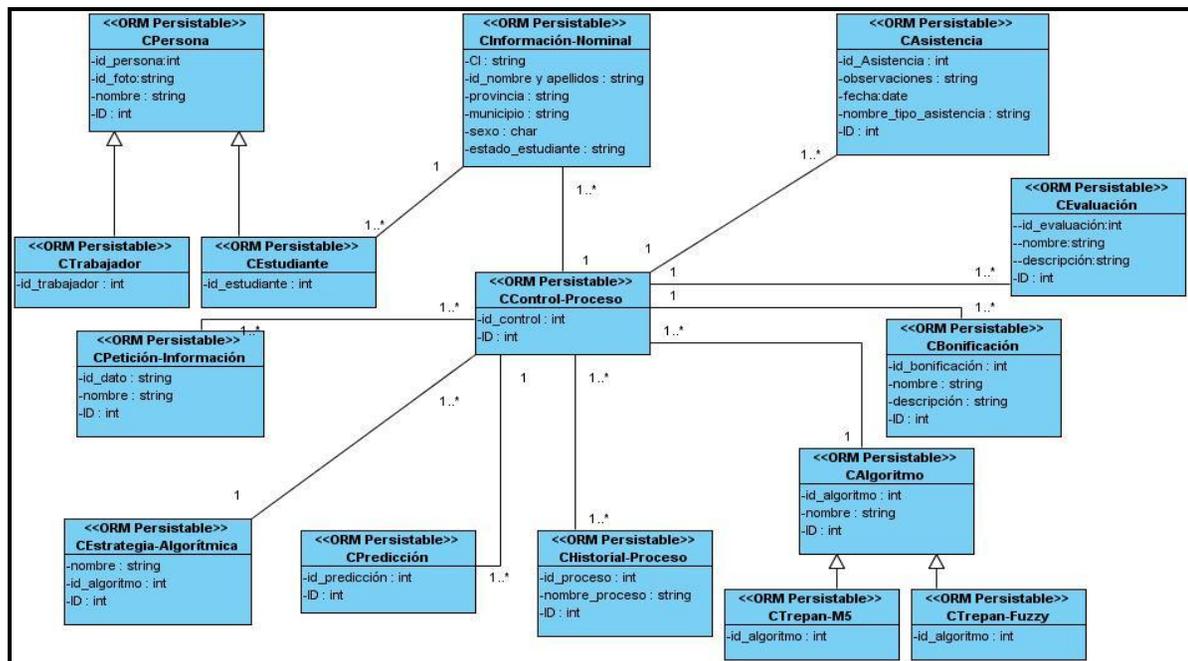


Fig. 13 Diagrama de Clases Persistentes

2.15.1 Descripción de las clases persistentes

Luego de ilustrar el diagrama de clases persistentes se procede a describir cada una de las clases, donde se detallan los atributos correspondientes a cada clase, el tipo y una breve descripción de los mismos. ([Ver Anexo 1](#)).

2.16 Diseño de la Base de Datos.

Para el sistema que se está diseñando se propone la implementación de una base de datos centralizada. A continuación se presenta el diseño de la base de datos propuesta:

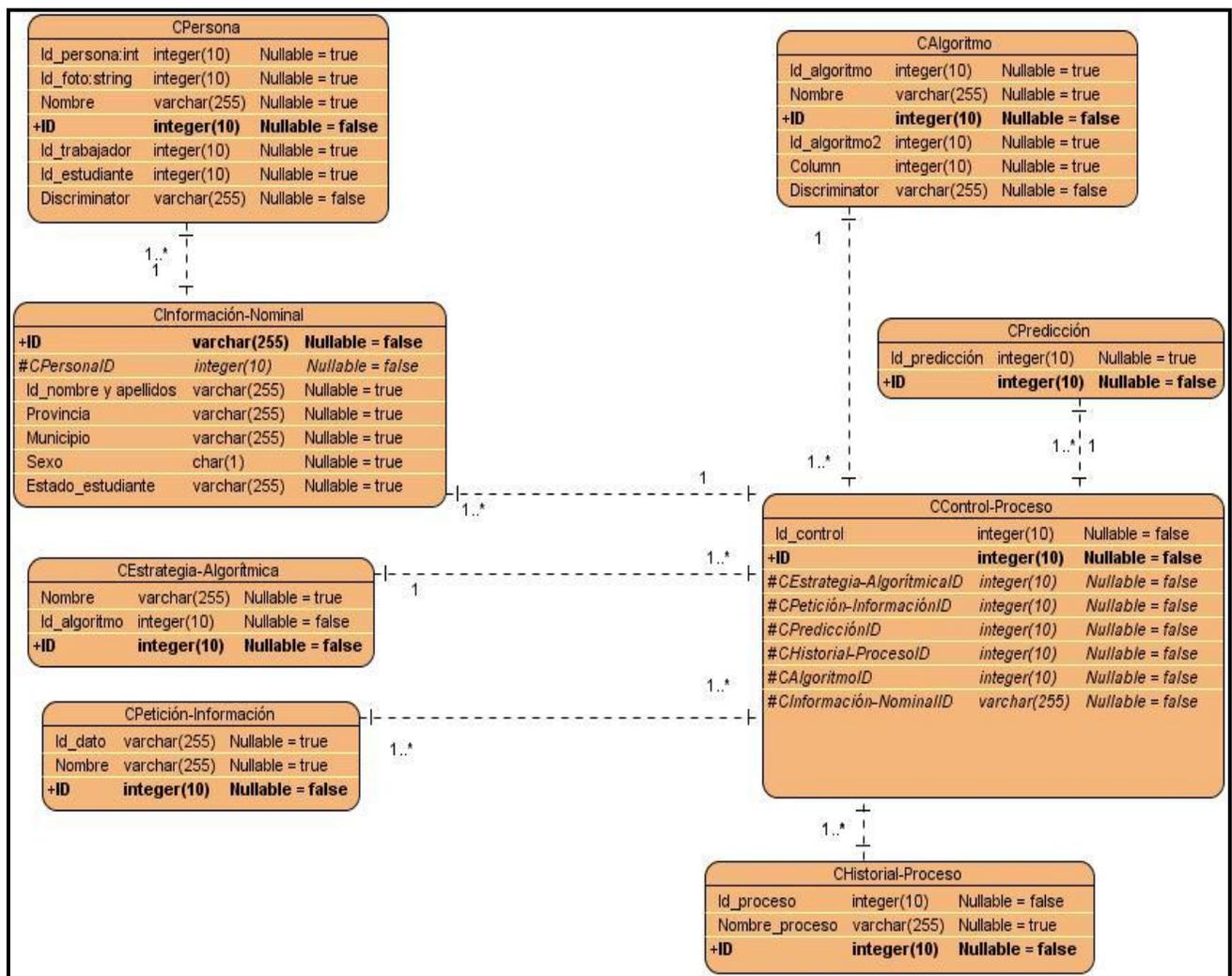
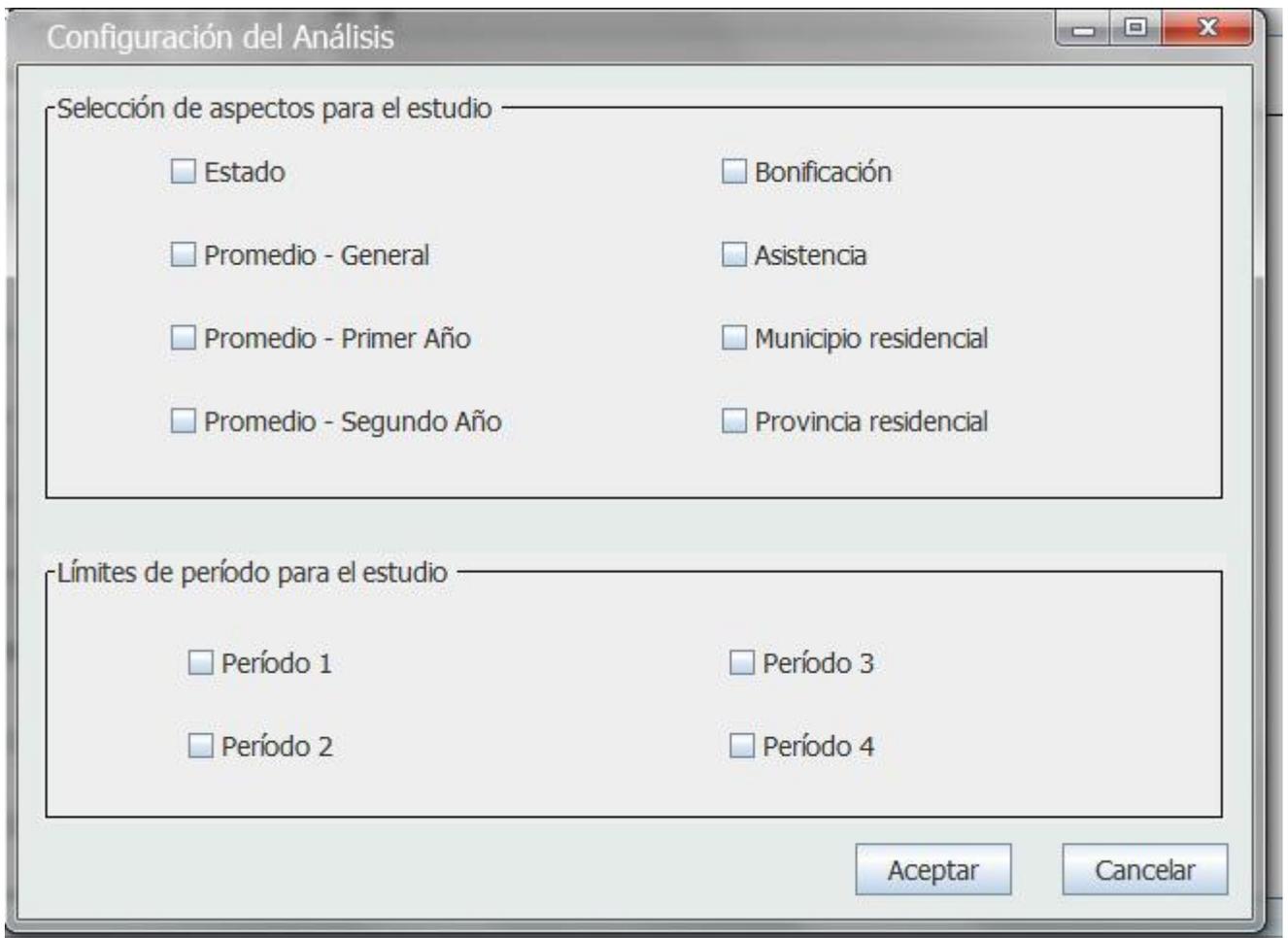


Fig. 14 Modelo de Datos

2.17 Prototipo de Interfaz de Usuario.



2.18 Conclusiones.

Se ha comprendido mejor las características del sistema a automatizar, teniendo como resultado el modelo del negocio y los actores que intervienen en este. Se realizó la captura de requisitos y se hizo una propuesta del sistema según las necesidades de los usuarios. Además se obtuvo el diagrama de casos de uso del sistema y la descripción de cada caso de uso. Al terminar este flujo de trabajo se puede comenzar el análisis y diseño a partir de los artefactos obtenidos. Se mostraron los diagramas de clases del diseño, así como el diagrama de secuencia correspondiente a cada caso de uso del sistema, se concedió el diagrama de clases persistentes y el modelo de datos. Se describieron las tablas del diagrama de clases persistentes y finalmente se ilustró el diagrama de despliegue.

CAPITULO 3: MÉTRICAS Y ESTUDIO DE FACTIBILIDAD.

3.1 Métricas para la evaluación del diseño.

Cuando los modelos de diseño aumentan en tamaño y complejidad resulta beneficioso aplicar métricas para verificar la calidad y objetividad del mismo. La misma busca comprobar, de manera cuantitativa, en qué grado el software posee las distintas características que definen la calidad de un producto de software, de ahí que su utilización sea una fase importante dentro de la evaluación del diseño. Entre las referenciadas por Pressman, se han seleccionado las que se aplican con mayor facilidad dentro del sistema en cuestión, considerando que este estudio brinda un modelo sencillo de implementar y que a su vez cubre los principales atributos de calidad de software. Lo referido al aseguramiento de la calidad es la principal razón por la que se conciben estas métricas.

La valoración de las actividades realizadas dentro del modelo de diseño se ha realizado con la aplicación de las siguientes métricas:

- Tamaño operacional de clase (TOC).

Esta métrica expresa el número de métodos asignados a una clase. El impacto que tiene su aplicación en los atributos de calidad mencionados se evidencia mayormente en la responsabilidad, complejidad y reutilización. La relación está expresada porque el aumento de la cantidad de métodos de la clase es directamente proporcional a la responsabilidad y complejidad de la clase, lo que supone un aumento de los mencionados indicadores. De igual manera, se traduce en una reducción de la reutilización que pueda tener la misma.

- Relaciones entre clases (RC)

Métrica que está dada por el número de relaciones de uso de una clase con otras. El aumento de las RC se traduce en un aumento del acoplamiento, complejidad de mantenimiento, cantidad de pruebas que requiere la clase; al tiempo que, de la misma forma que la métrica anterior, implica una reducción en la reutilización de la clase.

-Patrones

Bajo Acoplamiento:

Significa que una clase no depende de muchas clases. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas.

-Alta Cohesión:

La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Luego de efectuar los cálculos para un:

Total de clases	15
Promedio de procedimientos	3,666666667

Cantidad de procedimientos por clases

No	Nombre	Cantidad de Procedimientos
1	CPersona	4
2	CTrabajador	4
3	CEstudiente	6
4	CInformación-Nominal	4
5	CAsistencia	2
6	CEvaluación	2
7	CControl-Proceso	5
8	CPetición-Información	4
9	CEstrategia-Algorítmica	3
10	CPredicción	8
11	CHistorial-Proceso	6

Capítulo 3: Métricas y Estudio de Factibilidad

12	CAlgoritmo	2
13	CTrepan-M5	2
14	CTrepan-Fuzzy	2
15	CBonificación	1

Se obtuvieron las siguientes gráficas para la Responsabilidad, Complejidad y Reutilización, utilizando las siguientes tablas para obtener las categorías:

	Categoría	Criterio
Responsabilidad	Baja	\leq Prom.
	Media	Entre Prom. y 2^* Prom.
	Alta	$> 2^*$ Prom.
Complejidad implementación	Baja	\leq Prom.
	Media	Entre Prom. y 2^* Prom.
	Alta	$> 2^*$ Prom.
Reutilización	Baja	$> 2^*$ Prom.
	Media	Entre Prom. y 2^* Prom.
	Alta	\leq Prom.

La tabla “Responsabilidad” nos da una medida de los valores admisibles que deben tener las responsabilidades de las clases, la misma está asociada a la métrica Tamaño operacional de clases (TOC), de lo que se entiende que no se deben sobrecargar las clases en cuanto a operaciones.

La tabla “complejidad de implementación” también nos muestra el rango de valores que son admisibles para este parámetro, la misma esta dentro de la métrica (TOC) y nos dice que a medida que aumentan las operaciones aumenta su complejidad.

La tabla “reutilización” expresa que los valores de reutilización no deben ser bajos, esto se logra si las clases no son altamente complejas y presentan baja responsabilidad.

Responsabilidad y Complejidad de las clases.

Responsabilidad	Cantidad de clases	Promedio
Baja	15	28,84615385
Media	0	0
Alta	0	0

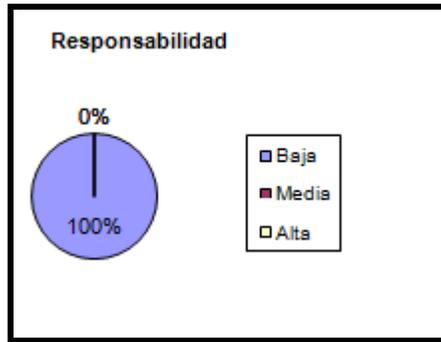


Fig. 15 Responsabilidad de las clases

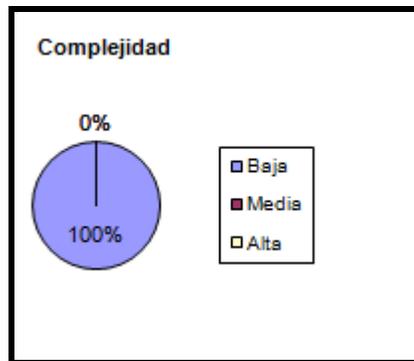


Fig. 16 Complejidad de las clases

Esta tabla representa que la cantidad de clases que se tenga tienen que estar en el nivel bajo o medio pero no el alto. Las dos figuras expresan visualmente que el 100% de las clases cuenta con una responsabilidad y una complejidad baja que son los valores deseados para dichos parámetros.

Reutilización de las clases.

Responsabilidad	Cantidad de clases	Promedio
Baja	15	28,84615385
Media	0	0
Alta	0	0

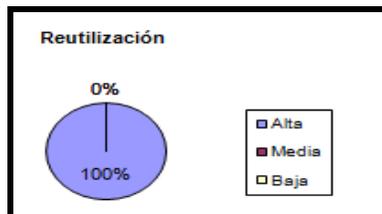


Fig. 17 Reutilización de las clases

Capítulo 3: Métricas y Estudio de Factibilidad

Todas nuestras clases son reutilizables ya que están entre Media y Alta, en la figura se muestra que hay un 100% de reutilización alta.

Cuando la responsabilidad (Figura 15) de las clases entre media y baja es mayor que el 80% se puede decir que cumple con el patrón de alta cohesión, además de que presentan una reutilización alta.

Para aplicar la métrica Relación entre Clases (RC) se tuvo en cuenta la cantidad de relaciones que tenían las clases, a partir del promedio de las relaciones y mediante un criterio se obtuvo la categoría (baja, media, alta y ninguna en caso del Acoplamiento) y (baja, media, alta) para la Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas.

Luego de efectuar los cálculos para un:

Total de clases	15
Promedio de asociaciones de uso	1,866666667

Reutilización de las clases.

No	Nombre	Cantidad de relaciones de uso
1	CPersona	2
2	CTrabajador	1
3	CEstudiante	2
4	CInformación-Nominal	2
5	CAсистенencia	1
6	CEvaluación	1
7	CControl-Proceso	9
8	CPetición-Información	1
9	CEstrategia-Algorítmica	1

10	CPredicción	1
11	CHistorial-Proceso	1
12	CAgoritmo	3
13	CTrepan-M5	1
14	CTrepan-Fuzzy	1
15	CBonificación	1

Tabla 9 Reutilización de las clases

Acoplamiento de las clases.

Acoplamiento	Cantidad de clases	Promedio
Ninguno	15	19,23076923
Bajo	10	12,82051282
Medio	3	3,846153846
Alto	2	2,564102564

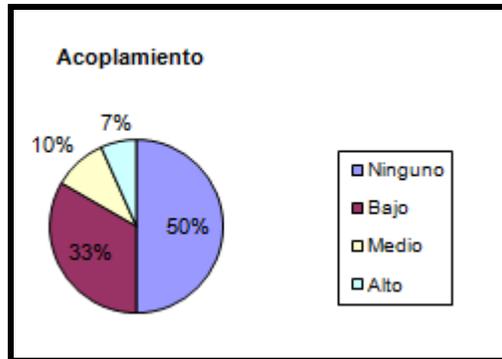


Figura 18: Acoplamiento de las clases.

Para un total de 15 clases se encuentran 10 con acoplamiento bajo y 5 con acoplamiento medio, en correspondencia con el patrón de bajo acoplamiento, como se muestra en la figura existe un 33% de clases que tienen medio acoplamiento y un 67% de clases de bajo acoplamiento.

Reutilización de las clases.

Reutilización	Cantidad de clases	Promedio
Baja	15	19,23076923
Media	4	5,128205128
Alta	10	12,82051282

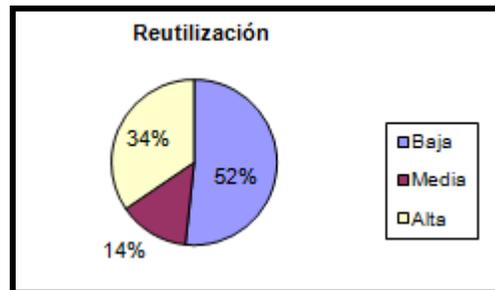


Fig. 19 Representación de la reutilización

Todas las clases son reutilizables, se encuentran entre Media y Alta para un total de 15 clases, como se muestra en la figura existe un 67% de alta reutilización y un 33% de media reutilización dando cumplimiento al patrón bajo acoplamiento.

3.2 *Estudio de Factibilidad.*

Después de definir la problemática presente y establecer las causas que ameritan de un nuevo sistema, es pertinente realizar un estudio de factibilidad para determinar la infraestructura tecnológica y la capacidad que implica la implantación del sistema en cuestión, así como los costos, beneficios y el grado de aceptación que la propuesta genera en la institución. Este análisis permitió determinar las posibilidades de diseñar el sistema propuesto y su puesta en marcha en la UCI, los aspectos tomados en cuenta para este estudio fueron clasificados en tres áreas, las cuales se describen a continuación:

3.2.1 *Factibilidad Técnica*

La Factibilidad Técnica consistió en realizar una evaluación de las tecnologías existentes en la universidad, este estudio estuvo destinado a recolectar información sobre los componentes técnicos con que cuenta la organización y la posibilidad de hacer uso de los mismos en el desarrollo e implementación del sistema propuesto y de ser necesario, los requerimientos tecnológicos que deben ser adquiridos para el desarrollo y puesta en marcha del sistema en cuestión.

De acuerdo a la tecnología necesaria para la implantación del sistema se evaluó en dos enfoques: Hardware y Software.

En cuanto a Hardware en el servidor debe estar instalado el sistema propuesto el cual debe cumplir con los siguientes requerimientos mínimos:

- Procesador Pentium 166 MHz
- Tarjeta Madre
- 64 MB de Memoria RAM
- Disco Duro de 5 GB
- Unidad de CD-ROM
- Tarjeta de Red
- Tarjeta de Video
- Unidad de Disco 31/2
- Monitor
- Teclado

-Mouse

-Unidad de protección UPS.

Evaluando el hardware existente y tomando en cuenta la configuración mínima necesaria, la Universidad no requiere realizar inversión inicial para la adquisición de nuevos equipos, ni tampoco mejorar rendimiento de los equipos existentes, ya que los mismos satisfacen los requerimientos establecidos tanto para el desarrollo como para el funcionamiento del sistema propuesto.

En cuanto al software, la Universidad cuenta con todas las herramientas necesarias para la implementación del mismo lo cual no amerita inversión alguna para la adquisición de éstas.

3.2.2 Factibilidad Económica

La Factibilidad Técnica arroja la determinación de los recursos para desarrollar, implantar, y mantener en operación el sistema propuesto, haciendo una evaluación donde se puso de manifiesto el equilibrio existente entre los costos del sistema y los beneficios que se pueden derivar de este, lo cual permitió observar de una manera más precisa los beneficios del sistema propuesto.

Análisis Costos-Beneficios:

Dentro del análisis realizado en cuanto al costo de los recursos y mano de obra necesaria para el desarrollo del sistema propuesto se llegó a la conclusión que como la Universidad cuenta con los equipos y recursos técnicos necesarios para el desarrollo del sistema por tanto no fue requerido ningún tipo de inversión adicional en este aspecto.

Dentro de los costos del personal en la Universidad el equipo de desarrollo no generaría ninguna inversión, ya que por ser una propuesta elaborada como tesis de grado, el personal propuesto para impulsar el desarrollo del sistema serían estudiantes por lo que no implica ningún gasto alguno; aspecto que favorecería aún más el proyecto en cuestión. Dentro de la fuerza de trabajo también intervendrían trabajadores, los cuales reciben un salario pero al mismo tiempo serían un ejemplo de trabajo y guía para los estudiantes.

Beneficios Tangibles:

Los beneficios tangibles del sistema propuesto están dados por el ahorro y mantenimiento a los equipos empleados.

Beneficios Intangibles:

Entre los beneficios intangibles del sistema propuesto se pueden incluir:

- Optimización y aumento de la productividad del personal, repercutiendo en el desarrollo de la UCI.
- Control y seguimiento de las tareas asignadas que permite un mejor y más efectivo empleo de los recursos.
- Generar información más eficiente y confiable, que sirva de apoyo a la toma de decisiones.
- Realzar la imagen de la Universidad de las Ciencias Informáticas colocándola al alcance de la comunidad.

Después de analizar el Costo-Beneficio finalmente se arriba a la conclusión de que el desarrollo de la propuesta del sistema presenta ventajas para la Universidad de las Ciencias Informáticas, ya que la misma cuenta con los recursos técnicos necesarios (hardware y software) para la implementación y explotación del nuevo sistema propuesto; así como en la mano de obra propuesta para el desarrollo de la misma.

De igual manera el nuevo sistema propuesto traería mejoras significativas para el desenvolvimiento y manejo de nuestra información reduciendo en tiempo y eficiencia de nuestros datos, disminuyendo las cargas de trabajo a los usuarios, ya que la velocidad de procesamiento, veracidad y confiabilidad de los procesos y resultados serán los deseados. El beneficio más significativo que se adjudicaría la organización sería la información convirtiéndose de esta manera en una de las herramientas más poderosas y versátiles con que cuente la universidad.

Es muy importante destacar que en esta nueva era de la informática, mejor conocida como la “Era de la Información”, este recurso es la herramienta de más competitividad utilizada por las organizaciones, y en cualquier caso, el desarrollo de aplicaciones podría significar ahorro, tanto de tiempo como de dinero.

Con la puesta en marcha de este proyecto se logrará optimizar los procesos que involucra el almacenamiento de gran cantidad de volúmenes de información, reduciendo en recursos tanto materiales como humanos, permitiendo obtener una información segura y confiable para la toma de decisiones.

3.2.3 Factibilidad Operativa

La factibilidad operativa permite predecir, si se pondrá en marcha el sistema propuesto, aprovechando los beneficios que ofrece, a todos los usuarios involucrados con el mismo, ya sean los que interactúan directamente con éste como también aquellos que reciben la información producida por el sistema. Por otro lado el buen funcionamiento del sistema en cuestión siempre estará adjudicado a la capacidad de los encargados de dicha tarea. En el proceso de adiestramiento se detallarán los aspectos de actualización de conocimientos del personal para el manejo del sistema propuesto.

Con la finalidad de garantizar el buen funcionamiento del sistema y que este impactará de forma positiva a los usuarios, el mismo sería desarrollado en forma estándar presentando una interfaz amigable al usuario, lo que se traduce en una herramienta de fácil manejo y comprensión, contando al mismo tiempo con la opinión de los usuarios para cualquier modificación que se desee realizar al sistema en cualquier momento.

3.3 Conclusiones.

Se aplicaron métricas para verificar la calidad y objetividad de los modelos de diseño. La misma comprobó de manera cuantitativa, en qué grado el software posee las distintas características que definen la calidad de un producto de software, de ahí que su utilización fue una fase importante dentro de la evaluación del diseño. Se han seleccionado las que se aplican con mayor facilidad dentro del sistema en cuestión, considerando que este estudio brinda un modelo sencillo de implementar. Posteriormente se realizó un estudio de factibilidad, se definió la problemática presente para establecer las causas que ameritan el nuevo sistema. Después de haber realizado un estudio sobre costos, beneficios y el grado de aceptación se arriba a que es factible la implantación y puesta en marcha de la propuesta del nuevo sistema.

CONCLUSIONES

A partir de la investigación realizada se arriba a las siguientes conclusiones:

- En el mundo existe un gran número de universidades que cuentan con sistemas automatizados de gestión académica y otras investigan la posibilidad de producirlos por la importancia de los procesos docentes en la sociedad, así como empresas que han creado productos donde usen las redes neuronales conjuntamente con la minería de datos.
- La modelación del sistema realizada utilizando el Proceso Unificado de Desarrollo de Software, permitió una mejor comprensión de los requisitos de la aplicación y formalización de los mismos.
- Para lograr gestionar información de manera fácil, permitiendo la toma efectiva de decisiones, con mantenimiento simplificado se debe utilizar una herramienta que permita hacer predicciones futuras.
- La propuesta del sistema tiene funcionalidades adicionales y necesarias para establecer dentro de la Aplicación Akademos predicciones que permitan la toma de decisiones dentro de un período dado.

RECOMENDACIONES

- La implementación del sistema con el fin de explotar todas las funcionalidades diseñadas.
- Continuar el estudio sobre el tema de predicciones utilizando redes neuronales conjuntamente con la minería de datos para una mejor implementación.

BIBLIOGRAFÍA

1. **Bot, Romina Laura.** *Data Mining utilizando Redes Neuronales.* Buenos Aires : s.n., 2005.
2. Data Mining y Data Warehousing. [En línea] 4 de 2008. <http://www.scribd.com/doc/19855790/Mineria-de-Datos-y-Data-Warehouse>.
3. Redes Neuronales Artificiales. [En línea] 19 de 4 de 2001. <http://electronica.com.mx/neural/>.
4. **Jaime, Hector Gonzalez.** *Tutorial de RNA .* ITESO Mexico : s.n., .
5. *Minería de Datos.* 2002.
6. Apara. [En línea] www.aparasw.com.
7. Gestión Académica. [En línea] 2010. http://www.ing.puc.cl/esp/ingeneral/ingenieria_en_cifras/informes_gestion/gestion_academica_2003.html.
8. Alyuda. [En línea] 2008. <http://www.alayuda.com/spanish/neural-network-software.htm>.
9. **M.J.A, Berry and Linoff, G.S. s.l. : Wiley.** *Mastering Data Mining.* 2000.
10. Área de Gestión Académica . [En línea] <http://www.um.es/academic/>.
11. IBM Rational Unified Process (RUP). [En línea] <http://www-01.ibm.com/software/awdtools/rup/>.
12. **Lic. Mónica Flores, Mtra. María de Lourdes.** Rational Process Library. [En línea] <http://www-01.ibm.com/software/awdtools/rmc/library/>.

ANEXOS

Anexo 1: Descripción de las clases persistentes.

Descripción de la clase: CPersona

Nombre: CPersona	
Tipo de clase Entidad	
Atributo	Tipo
id persona	int
id_foto	string
nombre	string
ID	int
Descripción:	
La clase CPersona (clase padre) tiene toda la información de las clases CTrabajador y CEstudiante (clases hijas).	

Descripción de la clase: CInformación-Nominal

Nombre: CInformación-Nominal	
Tipo de clase Entidad	
Atributo	Tipo
CI	string
id_nombre y apellidos	string
provincia	string
municipio	string
sexo	char
estado_estudiante	string

Descripción:

La clase CInformación-Nominal tiene toda la información de los estudiantes.

Descripción de la clase: CAsistencia

Nombre: CAsistencia	
Tipo de clase Entidad	
Atributo	Tipo
Id_Asistencia	int
observaciones	string
fecha	date
nombre_tipo_asistencia	string
ID	int
Descripción:	
La clase CAsistencia tiene toda la información sobre la asistencia de los estudiantes.	

Descripción de la clase: CEvaluación

Nombre: CEvaluación	
Tipo de clase Entidad	
Atributo	Tipo
id_evaluación	int
nombre	string
descripción	string
ID	int
Descripción:	

La clase CEvaluación tiene toda la información sobre las evaluaciones de los estudiantes.

Descripción de la clase: CBonificación

Nombre: CBonificación	
Tipo de clase Entidad	
Atributo	Tipo
id_bonificación	int
nombre	string
descripción	string
ID	int
Descripción:	
La clase CBonificación tiene toda la información sobre las bonificaciones que los estudiantes adquieran.	

Descripción de la clase: CAlgoritmo

Nombre: CAlgoritmo	
Tipo de clase Entidad	
Atributo	Tipo
id_algoritmo	int
nombre	string
ID	int
Descripción:	
La clase CAlgoritmo (clase padre) tiene la información de las clases CTrepan-M5 y CTrepan-Fuzzy las cuales son las clases hijas.	

Descripción de la clase: CControl-Proceso

Nombre: CControl-Proceso	
Tipo de clase Entidad	
Atributo	Tipo
id_control	int
ID	int
Descripción: La clase CControl-Proceso es donde se procesa toda la información necesaria para obtener predicciones.	

Descripción de la clase: CPetición-Información

Nombre: CPetición-Información	
Tipo de clase Entidad	
Atributo	Tipo
id_dato	string
nombre	string
ID	int
Descripción: La clase CPetición-Información es donde se almacena toda la información necesaria para el proceso.	

Descripción de la clase: CEstrategia-Algorítmica

Nombre: CEstrategia-Algorítmica	
Tipo de clase Entidad	
Atributo	Tipo

id_algoritmo	string
nombre	string
ID	int
Descripción:	
La clase C Estrategia-Algorítmica es donde se analiza cual algoritmo utilizar para posteriormente pasar al formateo de esos datos para que el algoritmo los entienda.	

Descripción de la clase: CPredicción

Nombre: CPredicción	
Tipo de clase Entidad	
Atributo	Tipo
id_predicción	int
ID	int
Descripción:	
La clase CPredicción es donde se almacena los resultados obtenidos después de procesados obteniéndose de esta manera predicciones.	

Descripción de la clase: CHistorial-Proceso

Nombre: CHistorial-Proceso	
Tipo de clase Entidad	
Atributo	Tipo
id_proceso	int
nombre_proceso	string
ID	int
Descripción:	

La clase CHistorial-Proceso es donde se almacena todas las predicciones obtenidas hasta ese momento.

GLOSARIO DE TÉRMINOS

- ¹ **TREPAN FUZZY:** Para extraer árboles difusos a partir de una red neuronal entrenada.
- ² **TREPAN M5:** Método para obtener árboles concisos y con un alto grado de flexibilidad.
- ³ **Trepan:** Algoritmo para construir árboles de decisión particionando el espacio de forma recursiva.
- ⁴ **Redes neuronales:** Colección de nodos conectados con entradas, salidas y procesamiento en cada nodo.
- ⁵ **IA:** Es todo el conocimiento y se puede representar mediante símbolos.
- ⁶ **UCI:** Universidad de las Ciencias Informáticas.
- ⁷ **RUP:** Proceso Unificado de Desarrollo.
- ⁸ **UML:** Lenguaje Unificado de Modelado.
- ⁹ **ASP:** Framework para aplicaciones web desarrollado y comercializado por Microsoft.
- ¹⁰ **PHP:** *Hypertext Pre-processor*, lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.
- ¹¹ **JSP:** Java Server Pages, tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.
- ¹² **HTML:** *Lenguaje de Marcado de Hipertexto*, es el lenguaje de marcado predominante para la elaboración de páginas web.
- ¹³ **CGI:** *Common Gateway Interface*, permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.
- ¹⁴ **PostgreSQL:** Sistema de gestión de base de datos relacional orientada a objetos y libre.
- ¹⁵ **CPUs:** Proporcionan la característica fundamental de la computadora digital y son uno de los componentes necesarios encontrados en las computadoras.
- ¹⁶ **Rollback's:** Es una operación que devuelve a la base de datos a algún estado previo.
- ¹⁷ **Open Source:** Es el término con el que se conoce al software distribuido y desarrollado libremente.

- ¹⁸ **Frameworks:** Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos.
- ¹⁹ **ExtJS:** Es una biblioteca de Java Script para el desarrollo de aplicaciones web.
- ²⁰ **Symfony:** Framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características.
- ²¹ **MYSQL:** Sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.
- ²² **SQLite:** Sistema de gestión de bases de datos relacional.
- ²³ **Oracle:** Se considera como uno de los sistemas de bases de datos más completos.
- ²⁴ **MS SQL Server:** Sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.
- ²⁵ **Plugins:** Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.
- ²⁶ **MVC:** *Modelo Vista Controlador*, un patrón de diseño en ingeniería de software.
- ²⁷ **ORM:** *Mapeo objeto-relacional*, es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos.
- ²⁸ **AJAX:** Técnica de desarrollo web para crear aplicaciones interactivas.
- ²⁹ **DHTML:** Designa el conjunto de técnicas que permiten crear sitios web interactivos utilizando una combinación de lenguaje HTML estático.
- ³⁰ **DOM:** *Modelo de Objetos del Documento*, proporciona un conjunto estándar de objetos para representar documentos HTML y XML.
- ³¹ **API:** Conjunto de funciones y procedimientos o métodos, en la programación orientada a objetos.
- ³² **Firefox:** Es un navegador multiplataforma y está disponible en varias versiones de Microsoft Windows.
- ³³ **IE:** Es un navegador web desarrollado por Microsoft para el sistema operativo Microsoft Windows.
- ³⁴ **Safari:** Navegador web de código cerrado desarrollado por Apple.

- ³⁵ **Binding:** Es una “ligadura” o referencia a otro símbolo más largo y complicado, y que se usa frecuentemente
- ³⁶ **Pooling:** Permite el acceso público a bandas de frecuencias privadas creando reservas comunes de espectro.
- ³⁷ **Java Swing:** Swing es una biblioteca gráfica para Java.
- ³⁸ **LGPL:** Esta licencia permisiva se aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario.
- ³⁹ **GPL:** Es un documento que cede ciertos derechos al usuario, asume la forma de un contrato, por lo que usualmente se la denomina contrato de licencia.
- ⁴⁰ **Zend:** Es un framework de código abierto para desarrollar aplicaciones web.
- ⁴¹ **CodeIgniter:** Es un entorno de desarrollo abierto que permite crear webs dinámicas con PHP.
- ⁴² **XML:** Permite definir la gramática de lenguajes específicos (de la misma manera que HTML).
- ⁴³ **SGML:** Sirve para especificar las reglas de etiquetado de documentos y no impone en sí ningún conjunto de etiquetas en especial.
- ⁴⁴ **BUGS:** Resultado de un fallo o deficiencia durante el proceso de creación de programas de ordenador o computadora (software).
- ⁴⁵ **SCRIPT:** Conjunto de instrucciones. Permiten la automatización de tareas, creando pequeñas utilidades.
- ⁴⁶ **Regresiones Lineales:** Método matemático que modeliza la relación entre una variable dependiente Y , las variables independientes X_i y un término aleatorio ϵ .
- ⁴⁷ **Árboles de Regresión:** Al igual que las regresiones lineales se utilizan para clasificar dataset con clases numéricas.
- ⁴⁸ **Árbol de Decisión:** Es un modelo de predicción utilizado en el ámbito de la inteligencia artificial.
- ⁴⁹ **Oráculo:** Es un programa que se usa en la prueba de software. Su principal función es indicar si un programa supera o no una prueba.
- ⁵⁰ **LINUX:** Es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente.