

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 2



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

Título: Herramienta informática de apoyo al proceso de elaboración de las estrategias educativas en la Facultad 2 empleando agrupamiento conceptual basado en distancia.

Autores:

Anniel Sánchez Verdecia

Jorge Raúl Gómez Lorenzo

Tutores:

Ing. Maidelis Milanés Luque

Msc. Yunia Reyes González

“La Habana, Julio del 2016”



“El fin de la educación no es hacer al hombre nulo, por el desdén o el acomodo imposible al país en que ha de vivir, sino prepararlo para vivir bueno y útil en él”

José Martí

Declaración de autoría

Declaramos ser autores del trabajo de diploma “Herramienta informática de apoyo al proceso de elaboración de las estrategias educativas en la Facultad 2 empleando agrupamiento conceptual basado en distancia” para optar por el título de Ingeniero en Ciencias Informáticas y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales con carácter exclusivo.

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

Anniel Sánchez Verdecia

Firma del Autor

Jorge Raúl Gómez Lorenzo

Firma del Autor

Ing. Maidelis Milanés Luque

Firma del Tutor

Msc. Yunia Reyes González

Firma del Tutor

Datos de contacto

Ing. Maidelis Milanés Luque: Graduada de Ingeniera en Ciencias Informáticas en el 2007, actualmente realizando investigaciones en el área de la Inteligencia Artificial, específicamente en la Minería de Datos (Algoritmos de agrupamiento). Profesora de Inteligencia Artificial, Reconocimiento de Patrones y Redes Neuronales Artificiales.

Correo electrónico: mmilanes@uci.cu

Msc. Yunia Reyes González: Graduada de Ingeniero en Ciencias Informáticas en el 2008, profesor de Inteligencia Artificial y Reconocimiento de Patrones. Máster desde el año 2014 en la temática Reconocimiento de Patrones con agrupamiento conceptual.

Correo electrónico: yrglez@uci.cu

Agradecimientos

Después de muchos años de estudio y esfuerzo, hoy me gradúo de Ingeniero en Ciencias informáticas. Para hacer realidad este sueño muchas personas me ayudaron, apoyaron y depositaron su confianza en mí. Hoy tengo la oportunidad de hacerles saber cuan agradecido estoy.

A mis abuelos que aunque no los tenga hoy físicamente conmigo los llevo dentro cada día que pasa, especialmente a mi abuelo Nicolás que lo perdí meses antes de entrar a la escuela y le prometí que terminaría por él, hoy estoy cumpliendo mi promesa, abue este logro también es tuyo.

A mis padres por todo su esfuerzo y sacrificio, por su cariño, su apoyo, gracias por estar siempre cuando los necesité. Quiero que sepan que los quiero y los admiro mucho y este logro es de ustedes también, nunca les podré estar lo suficientemente agradecido por todo lo que han hecho por mí.

A mi hermana, esa persona difícil de amar, pero cuando te falta sientes un vacío infinito. Por ser un ejemplo a seguir, por ser más que eso, por ser mi amiga, mi madre, mi compañera, mi guía, por ser esa persona que busco siempre para refugiarme en ella. Gracias tita por existir, te quiero mucho, no me imagino esta vida sin tí.

A Yenny por estar siempre conmigo, por tu amor, tu comprensión, por darme tu apoyo incondicional, siendo más que mi novia mi amiga. Gracias por estar ahí cuando lo necesité y no dejarme solo en ningún momento, aconsejándome y dándome toda tu ayuda en especial en estos últimos meses que han sido muy duros para mí. Por ser la madre de la alegría más grande que tengo, esa criatura que me hizo seguir adelante muchas veces cuando pensaba que no terminaba.

Agradecimientos

A mis tíos José, Nancy, Juanita, Matí, Ivon por esas personas en las que he podido apoyarme sin tener que dar nunca nada a cambio, por ser otros padres para mí.

A Tere por ser esa persona que sin llevar mi sangre es más que una familia y a su hijo Camilo que aunque hoy no me acompaña es mi hermano.

A mi primo Alián por ser como un hermano aunque en los últimos tiempos no hemos podido estar tan juntos como quisiera por situaciones que nos ha puesto la vida.

A Yaritza esa amiga especial que me ha acompañado durante estos 5 años y aunque hemos pasado por momentos difíciles siempre se ha mantenido a mi lado, por ser esa persona en la que me he podido apoyar siempre, te quiero mucho.

Al Zurdo que desde que nos conocimos ha crecido una amistad que sé que durará por mucho tiempo, por ser mi hermano, por querer ser el padrino de mi hijo y que sea como él dice que lo quiera como si fuese su hijo.

A Magdiel, otro hermanito mío, que aunque no nos conocimos desde 1er año hemos vivido juntos buenos y malos momentos pero siempre apoyándonos el uno al otro como si nos conociéramos desde niños.

A Raciél que lo conozco hace poco pero hemos hecho una amistad que ojalá y dure muchos años, también te quiero.

A mi compañero de tesis Jorge Raúl, que además de eso ha sido un amigo mío durante todos estos años y a su familia por ser tan atenta en aquellos días que estuvimos a toda hora en su casa.

Agradecimientos

En general, a todos esos amigos que me apoyaron desde el principio y los que fui conociendo en el camino, como Leonel, Ariel, Leandro, Suinny, Joel, Rafael, y todos los que me quedan por mencionar.

Anniel Sánchez Verdecia

Mi mamá como toda buena madre, ayuda a su hijo cualquiera sea la circunstancia, pero en mi caso nunca falta antes hacerme responsable de toda culpa, a la par del mejor de los regaños, incluso si antes le demostré mi inocencia. Luego, sin descanso, pelea junto a mí hasta la muerte, así sea en total desventaja. Le agradezco, por tanto, su dedicación y persistencia a lo largo de mi vida, en la crianza que me ha permitido ser ahora el hombre que soy.

A mi padre por ser incondicional. Papá no sabe decirme NO: por ejemplo, cuando no alcanzo a tomar la guagua de la UCI y tengo que virar corriendo a la casa y despertarlo a la 6:30 de la mañana para que me lleve en el carro, siendo posible que él haya terminado de trabajar en la madrugada, sin decir nada se levanta y camino al parqueo, consecuente con el apuro, va saludando a cuanta persona conozca (o no). Gracias papá por saberte siempre en mi disposición y estar presente en todos mis problemas y felicidades.

Mi hermano es este tipo de persona difícil de amar, pero cuando te falta sientes un vacío infinito. Siempre ha sido para mí una guía por donde transitar. Nunca faltan por su parte las críticas constructivas y no tan constructivas, tomadas por mí de la mejor manera, y los halagos no tan frecuentes hacen que me sienta más orgulloso de lo que puedo serlo por mí mismo.

Agradecimientos

A mis abuelos por ser la parte más importante del apoyo a mis padres en mi formación y por darme todo su cariño y atención en cada momento que lo necesité. En particular resaltar el apoyo de mi abuelo Ysidro en esta tesis, lo que implicó para él mantenerse despierto trabajando y luchando contra mi vagancia; de lo contrario dudo haber terminado a tiempo. Gracias también a mi tía Adita por su ayuda, lo que representó desde otra área del conocimiento una mejora en la perspectiva del trabajo.

A mi familia por numerosa que sea, quiero agradecerle por ser tan unida, cariñosa y por responder siempre en la ayuda de cada uno de nosotros.

A Alfre. Porque sí.

A Rey (por saber cómo ser buen amigo; por discutir cuando incluso no nos damos cuenta que hablamos de exactamente lo mismo; por hacerme combinar el trabajo y el juego, casi más juego que trabajo, que realmente a veces lo necesitaba; por su compañía en el gimnasio), el Compa (que ha sido un amigo heredado de mi hermano y ha hecho que lo sienta mi amigo también; por sus chistes en los momentos menos apropiados, que te liberan del stress y te alejan siempre de lo triste; también por las maripositas chinas), a Alberto (por su aporte práctico en la tesis; por su compañía en las reuniones en “la casita”, ya sea por la necesidad de buscar apoyo en los momentos tristes o festejar con o sin motivos; y por su compañía en el Tun Tun).

Al resto de mis amigos, que de alguna manera se las ingenian para soportarme y estar cuando lo necesito. Gracias a Sol por las madrugadas de café y cigarro escuchando mis protestas producto del stress de la tesis, y por estar siempre al tanto del progreso del trabajo.

Jorge Raúl Gómez Lorenzo

Dedicatoria

A mis padres, mis abuelos, mi hermana, mi hijo, mi novia y a todas las personas que me quieren y siempre confiaron en mí brindándome su apoyo para que viera realizado mi sueño, quiero regalarles este momento y honrarlos por tanto amor y dedicación. Los quiero mucho.

Anniel Sánchez Verdecía

A mis padres y hermano.

Jorge Raúl Gómez Lorenzo

Resumen

La aparición de las Nuevas Tecnologías de la Información y las Comunicaciones (TIC) ha marcado un cambio significativo en la sociedad con respecto a la información. Las nuevas posibilidades en el acceso, almacenamiento y utilización de éstas han motivado una toma de conciencia sobre el valor de la información en el ámbito educacional, dando origen a la implementación de nuevas formas de representar el conocimiento por medios no tradicionales. A raíz de esta tendencia, la Universidad de las Ciencias Informáticas (UCI), desde sus inicios, ha estado a la vanguardia en la creación de herramientas informáticas.

El presente trabajo de diploma tiene como objetivo desarrollar una herramienta informática para el apoyo al proceso de elaboración de las estrategias educativas en la Facultad 2 utilizando agrupamiento conceptual basado en distancia. Surge ante la necesidad de la gran cantidad de estudiantes que conviven en la UCI, el volumen considerable de información que se archiva sobre ellos en particular y al desconocimiento que se tiene de las posibles clases que se puedan formar para la concreción de acciones específicas.

Para dar cumplimiento a los objetivos trazados se documentaron los resultados del estudio realizado con respecto a las soluciones similares existentes, y las herramientas y tecnologías utilizadas, luego se expusieron los aspectos necesarios para la implementación de la aplicación mediante los flujos de trabajo implementación y pruebas que propone la metodología de desarrollo XP. Después de realizadas las pruebas se obtuvo una aplicación que posibilita el agrupamiento adecuado de estudiantes.

Palabras claves: agrupamiento, agrupamiento conceptual, distancia, herramienta.

Índice

Introducción	1
Capítulo 1. Fundamentación Teórica	8
Introducción.....	8
1.1 Conceptos asociados	8
1.1.1 Área pedagógica.....	8
1.1.2 Área informática.....	10
1.2 Herramientas informáticas afines	18
1.2.1 Herramientas internacionales.....	18
1.2.2 Herramientas nacionales	19
1.3 Metodología, lenguaje de programación y herramientas utilizadas.....	22
1.3.1 Herramientas	22
1.3.2 Lenguajes	24
1.3.3 Framework de desarrollo	26
1.3.4 Metodología XP	27
Conclusiones.....	28
Capítulo 2. Propuesta de solución	30
Introducción.....	30
2.1 Propuesta de solución	30
2.1.1 Declaración de la base de casos	31
2.1.2 Funciones de comparación de rasgos.....	36
2.1.3 Función de distancia o semejanza	37
2.1.4 Umbral de distancia o semejanza	37
2.1.5 Generalización	38
2.1.6 Distancia de enlace.....	38

Índice

2.1.7 Algoritmo HDCC (Modificado)	38
2.2 Funcionalidades del sistema	39
2.3 Propiedades del producto.....	40
2.3.1 Usabilidad	40
2.3.2 Interfaz.....	40
2.3.3 Rendimiento.....	40
2.3.4 Restricciones de diseño	41
2.4 Fase de Exploración.....	41
2.4.1 Historias de usuarios	41
2.5 Fase de Planificación	45
2.5.1 Estimación de esfuerzos por Historias de Usuario	45
2.5.2 Plan de Iteraciones	46
Conclusiones.....	48
Capítulo 3. Diseño, implementación y prueba.....	49
Introducción.....	49
3.1 Arquitectura de software.....	49
3.1.1 Arquitectura Cliente-Servidor	49
3.2 Fase de diseño.....	50
3.2.1 Patrón Arquitectónico.....	50
3.2.2 Patrones de diseño	52
3.2.3 Tarjeta Clase –Responsabilidad –Colaborador (CRC)	54
3.3 Modelo físico de la base de datos	56
3.4 Implementación	58
3.5 Tareas de ingeniería	58
3.6 Pruebas.....	61
3.6.1 Pruebas de Caja Blanca	61
3.6.2 Prueba de Rendimiento	64

Índice

Conclusiones.....	66
Conclusiones Generales.....	67
Recomendaciones.....	69
Referencias.....	70

Índice de Figuras

Índice de figuras

Figura 1: Características del diagnóstico	9
Figura 2: Aspectos más importantes en un diagnóstico	10
Figura 3: Una posible generalización de un conjunto de puntos en R^2	15
Figura 4: Cinco posibles generalizaciones de dos puntos en R^2	15
Figura 5: Una generalización de dos patrones en R^2	16
Figura 6: Un árbol de agrupamiento o dendrograma y sus n niveles.....	17
Figura 7: Arquitectura Cliente-Servidor	49
Figura 8: Patrón Modelo -Vista -Controlador	52
Figura 9: Modelo físico de la base de datos.....	57
Figura 10: Prueba de camino básico para el cálculo de la distancia de enlace	62
Figura 11: Prueba de camino básico para el cálculo del umbral de semejanza.....	63
Figura 12: Dependencia de la media de los tiempos de corrida contra cantidad de estudiantes almacenados base de datos	64
Figura 13: Decrecimiento de $\Delta n/n_0$ con el aumento del número de experimentos.....	65
Figura 14: Gráfico Log-Log para la media de los tiempos de ejecución que muestra un coeficiente de regresión lineal de 0.97	66

Índice de tablas

Tabla 1: Tabla comparativa de las herramientas afines	21
Tabla 2: Operacionalización de la variable diagnóstico.....	31
Tabla 3: Historia de usuario # 1: Gestionar estudiantes	41
Tabla 4: Historia de usuario # 2: Agrupar estudiantes.....	42
Tabla 5: Historia de usuario # 3: Mostrar patrón por agrupamiento.....	43
Tabla 6: Historia de usuario # 4: Mostrar estudiantes por agrupamiento.....	44
Tabla 7: Historia de Usuario # 5: Importar datos de estudiantes desde Excel	45
Tabla 8: Estimación de esfuerzo por Historia de Usuario	46
Tabla 9: Plan de duración de la iteración 1	47
Tabla 10: Plan de duración de la iteración 2	47
Tabla 11: Plan de duración de la iteración 3	47
Tabla 12: Tarjeta CRC: Patrón.....	54
Tabla 13: Tarjeta CRC: Grupo	54
Tabla 14: Tarjeta CRC: Principal	55
Tabla 15: Tarjeta CRC: Visual	55
Tabla 16: Tarjeta CRC: Visual2	56
Tabla 17: Tarjeta CRC: Visual3	56
Tabla 18: Ejemplo de una Tarea de Ingeniería	59
Tabla 19: Tarea de Ingeniería # 1: Gestionar estudiantes.....	59
Tabla 20: Tarea de Ingeniería # 2: Agrupar estudiantes	59
Tabla 21: Tarea de Ingeniería # 3: Mostrar patrón por agrupamiento	60
Tabla 22: Tarea de Ingeniería # 4: Mostrar estudiantes por agrupamiento	60
Tabla 23: Tarea de ingeniería # 5: Importar datos de estudiantes desde fichero .csv	61

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) son en la actualidad temas de gran importancia que conforman un conjunto de recursos que facilitan la transmisión de la información, el manejo de los servicios, las herramientas, las tecnologías y la educación, obteniendo un lugar muy importante en la vida humana contemporánea. El uso de las TIC está en constante desarrollo en la esfera de la educación, brindando las herramientas necesarias, tanto para el fortalecimiento del proceso de enseñanza-aprendizaje, como para la creación de herramientas que complementen dicho proceso.

En los últimos años en Cuba se ha evidenciado un crecimiento en el uso de las TIC como recurso indispensable en el desarrollo económico, político y social del país. Con la creación de la Universidad de las Ciencias Informáticas (UCI) en el año 2002 se muestra un ejemplo de los numerosos esfuerzos que se están realizando en aras de potenciar la industria del software. Entre las misiones de la UCI, se encuentra la de formar profesionales competentes, pero que además tengan inculcados los valores reflejados en el modelo del profesional, entre los que se destacan la honestidad, responsabilidad, compromiso y sentido de pertenencia.

Para contribuir al cumplimiento con éxito de la formación integral de los futuros profesionales, se necesita de la elaboración y puesta en práctica de una estrategia educativa, y para ello es imprescindible la aplicación de un diagnóstico que brinde la información requerida de cada estudiante.

La UCI está actualmente compuesta por 6 facultades. Para la elaboración de las estrategias educativas en la Facultad 2, se realiza un diagnóstico inicial, del que los docentes extraen las características del estudiante y los posibles problemas que presentan, a partir del análisis manual del mismo por parte de los profesores guías, se clasifican esos problemas en 3 dimensiones y se proponen una serie de acciones para mitigarlos o eliminarlos.

Para la caracterización del estudiante solo se tienen en cuenta una serie de aspectos que se identifican por el docente, sin embargo, en los archivos digitales con que cuenta la UCI existen otros aspectos que pueden ser de interés para el tema que se aborda. Además el análisis de la información y el desglose de las posibles características comunes se

Introducción

realizan de manera manual, y no se tiene en cuenta los diferentes grupos con características similares que se pueden identificar.

Por tanto, se hace cada día más necesario elevar la calidad de este proceder pedagógico y para ello, su planificación tiene un rol muy importante.

Para cumplir con este fin se necesita tener en primer lugar una caracterización exhaustiva de cada estudiante para aplicar una atención grupal e individualizada a los mismos. Preguntas como, ¿qué características debe tener el diagnóstico y sus instrumentos para recoger la información sobre los estudiantes, necesarias para la elaboración de la estrategia educativa?, ¿qué dimensiones e indicadores deberán tenerse en cuenta y cómo categorizarlos? Esto conlleva un trabajo muy minucioso del docente y procesar mucha información, que por una u otra vía pueda adquirirse.

En la actualidad los docentes de la Facultad 2 requieren de un instrumento informático que les permita una mayor organización de la información que se requiere para categorizar a los estudiantes y agruparlos por características semejantes, lo que le permite al docente extraer las fortalezas y debilidades y por tanto la agilización de la elaboración de las estrategias educativas que se diseñen. Un uso efectivo de este instrumento podrá llevar a los docentes y directivos a una mejor orientación y desarrollo del proceso docente educativo, en las direcciones instructivas, formativas y desarrolladoras, que contribuyan a la formación integral de la personalidad de los educandos.

En correspondencia con esta situación problemática los autores de este trabajo se han planteado el siguiente **problema de investigación**:

¿Cómo contribuir desde la informática a mejorar el proceso de elaboración de las estrategias educativas en la Facultad 2 de la UCI?

Partiendo de este problema se define como **objeto de estudio** los algoritmos de agrupamiento y su vinculación con el trabajo educativo en la Facultad 2 de la UCI.

Como **campo de acción** se tienen los algoritmos de agrupamiento conceptual y su vinculación con el proceso de elaboración de estrategias educativas.

Introducción

Para dar solución al problema planteado se ha propuesto como **objetivo general**: Desarrollar una herramienta informática para el apoyo al proceso de elaboración de las estrategias educativas en la Facultad 2 utilizando el agrupamiento conceptual.

Para dar solución al objetivo general se formularon los siguientes **objetivos específicos**:

1. Fundamentar los referentes teóricos necesarios para la solución del problema planteado.
2. Diseñar la propuesta de solución que responda a la problemática presentada.
3. Validar la solución mediante las pruebas de software.

A partir de la definición de los objetivos específicos se plantean las siguientes **tareas de la Investigación**:

- Fundamentación de los referentes teóricos.
- Identificación del rol del diagnóstico en la estrategia educativa.
- Precisión de los requisitos de los algoritmos de agrupamiento conceptual, que satisfagan el objetivo de la herramienta concebida.
- Selección del algoritmo de agrupamiento conceptual a utilizar.
- Determinación de las herramientas informáticas existentes nacionales e internacionales, que se asemejen a la solución de la situación problemática planteada.
- Selección de las herramientas, metodología y tecnologías para el desarrollo de la propuesta de solución.
- Elaboración de la propuesta de solución a partir de las herramientas, metodología y tecnologías seleccionadas.
- Implementación de la solución.
- Ejecución de pruebas al software para identificar sus aciertos y deficiencias.
- Corrección de las deficiencias detectadas.

Introducción

Para el desarrollo adecuado de la investigación se utilizaron una serie de **métodos de investigación** teóricos y empíricos que a continuación se describen:

- **Métodos teóricos**

Analítico - sintético: permitió analizar y extraer las teorías y los documentos relativos al objeto de la investigación, proporcionando de esta manera la obtención de elementos del conocimiento fundamentales, relacionados con el agrupamiento conceptual como aspecto fundamental de la misma.

Inductivo-deductivo: permitió a partir de casos particulares llegar a conclusiones parciales y generales.

Modelación: este método permitió la creación de modelos mediante propuestas, alternativas y estrategias.

- **Métodos empíricos**

Entrevista no estructurada: este método se realizó mediante conversaciones que se llevaron a cabo con especialistas en el tema de la investigación. De acuerdo a la fluidez de la conversación la entrevista se fue desarrollando, y en correspondencia a las respuestas que ofrecían estos especialistas se fueron formulando las preguntas y así profundizando en los contenidos.

Encuesta opinática: mediante la utilización de este método, se aplicó a profesores de la UCI y expertos en el área educativa, un cuestionario (Anexos 1 y 2) que posibilitó la concreción de las dimensiones y los indicadores a tener en cuenta en el diagnóstico, para así determinar los ítems a utilizar en su valoración.

Se ha asumido como **variable dependiente** el diagnóstico, y esta se ha operacionalizado. Se analizaron las dimensiones académico, socio-político y extensionista utilizadas por la UCI para diagnósticos de este tipo. Por el volumen de información que contienen estas dimensiones se consideraron subvariables de esta investigación.

Se ha consultado bibliografía de distintos especialistas que tratan el tema, y se han analizado los distintos criterios que plantean sobre lo que debe tenerse en cuenta para la elaboración del diagnóstico. Estos criterios devienen en dimensiones de las diferentes subvariables que se tratan en esta investigación y ellas en distintos indicadores. Cada

Introducción

indicador se categoriza por diferentes ítems, que dan un criterio evaluativo de su estado (1).

Estas dimensiones son: motivos e intereses, académico, social, valores, normas de comportamiento, nivel de vida, relación familiar y datos generales. En el capítulo 2 “Propuesta de solución” aparece la operacionalización de la variable con todos sus aspectos.

Como análisis exploratorio y complemento en la realización del estudio del objeto de esta investigación, se decidió realizar la obtención de información sobre las dimensiones e indicadores a precisar para establecer una estrategia didáctica de la enseñanza. A veces para estudios exploratorios el muestreo probabilístico no resulta el más adecuado ya que es extremadamente costoso, en casos como éste, se acude a métodos no probabilísticos.

Es sabido que dichos métodos no posibilitan la generalización de los resultados obtenidos, pues se tiene la certeza de que no tiene la representatividad para generalizar los resultados, ya que los sujetos de la población no tienen la misma probabilidad de ser elegidos y en general se seleccionan los sujetos siguiendo determinados criterios. Este estudio exploratorio es considerado como un primer acercamiento a un problema y se utiliza cuando éste no ha sido aún abordado o no ha sido suficientemente estudiado.

Dentro de los métodos no probabilísticos se encuentra el muestreo opinático o intencional, que se basa en la opinión del investigador para establecer una muestra de sujetos que dé respuesta a determinados objetivos de estudio.

En el muestreo opinático el investigador selecciona a los sujetos que reúnan un grupo de cualidades, no obedece a reglas fijas ni específicas. De antemano el número de unidades a seleccionar, acepta en principio, que este puede ser alterado a lo largo de la investigación de manera que puedan seleccionarse unidades de muestreo no previstas inicialmente para mejorar la calidad y riqueza de la información. Además puede interrumpirse cuando se entiende que se ha llegado a un punto de saturación por la cantidad de información recogida o por criterio del investigador que considere que los nuevos datos comienzan a ser repetitivos y dejan de aportar información novedosa (2).

Dentro del muestreo opinático se incluye el juicio o criterio de expertos que consiste en solicitar a una serie de personas sus opiniones con relación a un objeto, un instrumento, un material de enseñanza o su opinión respecto a un aspecto concreto. Presenta muchas

Introducción

ventajas entre las que se destaca la posibilidad de obtener una amplia calidad de las respuestas.

Es por lo anteriormente expuesto, que se decidió obtener información sobre los componentes del diagnóstico de los estudiantes, mediante la consulta a un grupo de expertos cuyos vínculos fueran directos con la educación. Se establecieron dos grupos: uno de expertos y otro de especialistas. Para el establecimiento del primero se tomaron en consideración las siguientes cualidades: categoría científica, categoría docente, grado científico, años experiencia docente, autoría de libros, tutoría de tesis, entre otros. En el segundo grupo se incluyeron profesores actualmente vinculados a la docencia en la UCI (3).

A ambos grupos se les aplicó un cuestionario estructurado con preguntas cerradas y la posibilidad de que incorporaran aspectos que consideraran importante incluir. El objetivo del cuestionario es la identificación de las dimensiones e indicadores a tener en cuenta para la elaboración del diagnóstico del estudiante, a utilizar en la elaboración de una estrategia educativa.

Una vez determinadas las dimensiones se les solicitó establecer la prioridad o grado de importancia que le considera a las mismas. En preguntas posteriores, para cada una de las dimensiones establecidas por los entrevistados fueron precisados los indicadores que consideran que las caracterizan.

Los resultados presentados en las tablas de salidas se representan porcentualmente para una mejor comprensión (Anexos 3 y 4).

Debido a la similitud que muestra la información recogida en las encuestas aplicadas tanto a expertos como a profesores, la interpretación de los resultados que se mostrará es común para ambas.

De las dimensiones presentadas todas se consideran aceptadas para su inclusión en el diagnóstico de caracterización de estudiantes. Las dimensiones Motivos e intereses y Académico son consideradas con el máximo grado de importancia y el resto fluctúa entre el segundo y el quinto grado de importancia.

La mayor parte de los indicadores son aceptados por todos los entrevistados como válidos en la caracterización de las dimensiones. Los indicadores que no fueron

Introducción

aceptados por unanimidad, aun así son incluidos en el diagnóstico por ser considerados por la mayoría.

Como un primer acercamiento se pueden considerar estos indicadores y dimensiones adecuados en el diagnóstico de los estudiantes.

La tesis está estructurada en 3 capítulos.

Capítulo 1: Fundamentación Teórica.

En este capítulo se plantean los elementos teóricos que sustentan el trabajo, tanto pedagógicos como informáticos, y se realizó un estudio del estado del arte. Además, se hace un profundo análisis de las principales tendencias y tecnologías actuales, con el objetivo de seleccionar las herramientas a utilizar para la solución del problema planteado.

Capítulo 2: Propuesta de solución.

Se describe la propuesta de solución partiendo de los contenidos relacionados con las características del diagnóstico pedagógico a aplicar, para la recogida de información y elaboración de la estrategia educativa. Se aborda lo referente a la herramienta informática con el análisis del sistema, las funcionalidades que debe cumplir la herramienta y las propiedades que el producto debe poseer. Se realiza una descripción de las historias de usuario que se proponen para dar respuesta al objetivo general, derivado de la situación problemática. Además, se realiza la estimación de esfuerzo por cada historia de usuario y el plan de duración de las iteraciones a llevar a cabo.

Capítulo 3: Diseño, implementación y pruebas.

En este capítulo se describen los aspectos relacionados con la implementación del sistema, se especifica su arquitectura y con ello, el patrón arquitectónico empleado. Se adapta el diseño al entorno de implementación y se describen las tareas de ingeniería. Se realizan las pruebas de software para comprobar la calidad del producto.

El resultado del trabajo tiene actualidad científica y significación práctica, pues mediante el proceder informático basado en la inteligencia artificial y los aspectos pedagógicos, se pone en manos de docentes y directivos de la UCI un instrumento de mucho valor informático y educativo. El trabajo es pertinente, pues con la herramienta se puede atender las diferencias individuales de los educandos, aspecto que demanda la sociedad cubana y su sistema educacional.

Capítulo 1. Fundamentación teórica

Capítulo 1. Fundamentación Teórica

Introducción

En el presente capítulo se abordan los principales conceptos relacionados con el dominio del problema. Se realiza un estudio del estado del arte de algunas herramientas nacionales e internacionales relacionadas con la propuesta de solución, para la actualización del contenido del tema en cuestión, tanto en el plano pedagógico como informático, que propicie un mejor análisis y posible aplicación, teniendo en cuenta las condiciones en que se realiza la investigación y el objetivo planteado. Además, se detallan las tendencias y tecnologías actuales, metodologías y herramientas para el desarrollo de la solución propuesta en el campo informático.

1.1 Conceptos asociados

En el tratamiento para la solución del problema de investigación y el cumplimiento de su objetivo, se han tenido que precisar y aplicar conceptos relacionados con las áreas pedagógica e informática. El centro de atención del trabajo realizado ha estado orientado hacia la elaboración de la herramienta informática para propiciar la elaboración de las estrategias educativas a seguir, por los docentes, en el desarrollo del proceso docente educativo.

1.1.1 Área pedagógica

El fin de la educación cubana va dirigido a la formación integral de la personalidad de los educandos. En el caso concreto de las universidades cubanas las características del egresado se describen en el modelo del profesional de cada una de las diferentes carreras.

Para contribuir al cumplimiento con éxito de la formación integral de los futuros profesionales se necesita en primer lugar la elaboración por el docente de una estrategia educativa que lo oriente en el cumplimiento de sus fines. En la bibliografía pedagógica consultada se evidencia la importancia de este concepto y aparecen diversas definiciones del mismo. Se ha asumido la planteada por la Dr. C. María Antonia Rodríguez del Castillo y la Lic. Alvarina Rodríguez Palacios, las que señalan "**Estrategia educativa**: es la proyección de un sistema de acciones a corto, mediano y largo plazos que permiten la transformación de los modos de actuación de los escolares para alcanzar en un tiempo

Capítulo 1. Fundamentación teórica

concreto los objetivos comprometidos con la formación, desarrollo y perfeccionamiento de sus facultades morales e intelectuales” (4), por considerarla la que más se ajusta a los intereses de esta tesis.

Para una adecuada elaboración y desarrollo de las estrategias educativas, como ya se ha dicho es necesario tener una caracterización exhaustiva de cada estudiante. Esto puede lograrse mediante la aplicación del denominado diagnóstico. En los últimos veinte años este concepto ha sido tratado por diversos pedagogos.

En Cuba especialistas del Instituto Central de Ciencias Pedagógicas y de las Universidades Pedagógicas han desarrollado una amplia teoría sobre este tema, teniendo en cuenta el fin de la educación cubana, que en palabras del Apóstol es “preparar al hombre para la vida”, mediante una formación integral de su personalidad.

Entre estos autores pueden mencionarse entre otros, Valle L., Rico P., Fiallo J., Silvestre, M. y Zilberstein J. Estos dos últimos Doctores en Ciencias Pedagógicas definen el **diagnóstico** como “*el estudio profundo del estado de un proceso o producto de carácter pedagógico, que posibilita la identificación de logros, dificultades, potencialidades y causas, en función de un objetivo determinado, en un momento dado, con el propósito de su transformación*” (5).

Existe bastante consenso en las características del diagnóstico. Las más comunes planteadas por los diferentes autores son (4):

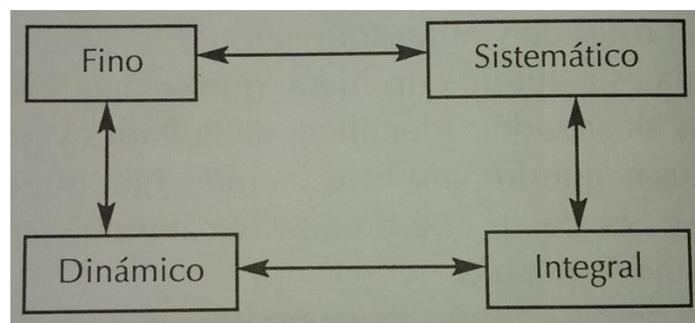


Figura 1: Características del diagnóstico

Cada uno de estos autores parte del planteamiento de un conjunto de aspectos que deben recogerse en la elaboración y aplicación del diagnóstico. Así por ejemplo, la Dra. Margarita Silvestre Oramas, en uno de sus libros (5), señala abordar el diagnóstico del estudiante de forma integral, recogiendo los aspectos siguientes:

Capítulo 1. Fundamentación teórica



Figura 2: Aspectos más importantes en un diagnóstico

Los aspectos generales planteados por estos autores han servido de base para la selección de los criterios que conforman la base de datos para elaborar la herramienta informática que se presenta en este trabajo.

Estos aspectos pueden reducirse o incrementarse dado las características del nivel de enseñanza de que se trate o la situación concreta en que se desarrolla el proceso formativo.

1.1.2 Área informática

Dadas las características y el nivel de información tan amplio que se requiere concretar en el área informática asociada al cumplimiento del objeto de investigación, se precisarán solamente las principales definiciones de conceptos y los elementos del conocimiento esenciales para la comprensión del trabajo realizado.

Reconocimientos de patrones: Es la zona del conocimiento (de carácter interdisciplinario) que se ocupa del desarrollo de teorías, métodos, técnicas, y dispositivos computacionales para la realización de procesos ingenieriles, computacionales y/o matemáticos, relacionados con objetos físicos y/o abstractos, que tienen el propósito de extraer la información que le permita establecer propiedades y/o vínculos de o entre conjuntos de dichos objetos sobre la base de los cuales se realiza una tarea de identificación o clasificación (6).

Capítulo 1. Fundamentación teórica

El reconocimiento de patrones concibe dos tipos de aprendizaje. Ellos son:

- **Aprendizaje supervisado:** Dado un universo de objetos y el conocimiento acerca de la existencia de ciertas clases con características (propiedades) de especial interés y una muestra de objetos que pertenecen a cada una de ellas, el problema consiste en determinar para cada uno de los objetos no clasificados las relaciones de pertenencia de los mismos con cada una de las clases (7).
- **Aprendizaje no supervisado:** El propósito es juntar (agrupar) los objetos según su analogía (parecido, semejanza, cercanía si se está hablando de un espacio de representación con distancia definida). En este sentido se pueden encontrar dos situaciones diferentes, a saber, el número de grupos es conocido previamente o no (7).

Como parte del aprendizaje supervisado tiene una gran importancia el término clasificación de los elementos de la base de conocimientos. Puede definirse este concepto como:

Clasificación: Permite dividir un conjunto de datos en grupos mutuamente excluyentes de tal manera que cada miembro de un grupo esté lo "más cercano" posible a otro, y grupos diferentes estén lo "más lejos" posible uno del otro, donde la distancia está medida con respecto a variable(s) específica(s), las cuales se están tratando de predecir (8).

En el caso del aprendizaje no supervisado resalta el término agrupamiento como un concepto determinante en esta tesis, pues el funcionamiento de la herramienta informática elaborada está basado en él. La definición asumida por los autores se recoge a continuación:

Agrupamiento: Es la clasificación de objetos dentro de diferentes grupos de forma automática, donde los objetos de un mismo grupo son similares entre ellos y disimilares con los objetos pertenecientes a otros grupos. La cantidad de grupos a formar puede estar previamente definida o no. De los objetos se conocen sus descripciones pero se desconocen las relaciones que existen entre ellos (9).

Para la realización de este agrupamiento se utilizan una serie de pasos, con un orden lógico determinado que se le conoce como algoritmo.

Capítulo 1. Fundamentación teórica

Algoritmo de agrupamiento: Un algoritmo de agrupamiento es un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio. Esos criterios son por lo general distancia o similitud. La cercanía se define en términos de una determinada función de distancia, como la euclídea, aunque existen otras más robustas o que permiten extenderla a variables discretas. La medida más utilizada para medir la similitud entre los casos es la matriz de semejanza entre los $n \times n$ casos (10).

Dado el objetivo de la investigación que plantea la elaboración de una herramienta informática para el uso del personal docente, se ha considerado necesario que esta, además de agrupar a los educandos, proporcione las características de los grupos creados, para una mejor atención grupal e individualizada de sus miembros. Con ello, se proporciona al docente la identificación de dificultades, potencialidades y causas, en función del objetivo formativo predeterminado, que le permita la transformación de los educandos. Esto es darle un carácter formativo al agrupamiento. Por tal motivo hemos utilizado el concepto agrupamiento conceptual, el cual se define como:

Agrupamiento conceptual: El agrupamiento conceptual resuelve el problema de interpretación de los grupos formando grupos que pueden ser descritos por propiedades que involucran relaciones sobre un conjunto selecto de atributos (11). Al igual que en el caso del agrupamiento, en este se requiere el uso de los algoritmos de agrupamiento agregando la parte conceptual.

El algoritmo que se utiliza para la realización de esta herramienta, se considera necesario que debe cumplir con las siguientes características :

1. El hecho de ser un algoritmo cuyas respuestas no son dependientes del orden de presentación de los objetos permite que no se condicione ninguna estructuración en función del orden de los objetos.
2. Tiene la capacidad de manejar atributos cualitativos y cuantitativos mezclados. Esta característica es significativa, pues se presenta con frecuencia en las bases de casos construidas a partir de las bases de datos.
3. No requiere especificar el número de agrupamientos a priori ni semillas, lo que implica una gran ventaja, pues es deseable organizar la base de casos en grupos considerando las semejanzas entre los casos, sin imponer una determinada cantidad de grupos a formar.

Capítulo 1. Fundamentación teórica

4. Los conceptos que genera son fáciles de interpretar por los usuarios, pues se obtienen en función de las propias variables que describen los objetos y distinguen a los objetos de cada grupo. Esto facilita comprender las características fundamentales que describen los agrupamientos de casos y además, reduce el espacio de búsqueda de los casos más similares dado un nuevo problema, pues bastaría con comparar con los conceptos distintivos de cada grupo para determinar a cuál se parece más.

Por esta razón es que se asume el algoritmo HDCC (**H**ierarchical **D**istance-based **C**onceptual **C**lustering) como alternativa para dar posible solución a esta investigación, ya que cumple con todas las características descritas anteriormente, además de que este método constituye una vía práctica y general para integrar métodos jerárquicos, basados en distancias con algoritmos conceptuales.

Agrupamiento basado en distancia (11)

El agrupamiento es partir el grupo de entrada en pequeños grupos, para este particionamiento se usan distancias. Dichas distancias se utilizan como funciones de semejanza, que nos permiten cuantificar la similitud entre dos objetos. Estas funciones transforman pares de objetos en números reales. Cuanto menor es este número, más similares son los objetos. La idea de asignar distancias a pares de puntos es precisamente lo que da origen a los espacios métricos.

Definición 1 Un espacio métrico es un par (X,d) donde X es un conjunto no vacío y d es una función real llamada distancia o métrica definida sobre $X \times X$ que satisface las siguientes propiedades:

- i) $d(x,y) \geq 0 \forall x, y \in X$ and $d(x,y) = 0 \iff x = y$
- ii) $d(x,y) = d(y,x) \forall x, y \in X$
- iii) $d(x,z) \leq d(x,y) + d(y,z) \forall x, y, z \in X$

Es decir, i) las distancias son siempre positivas y el único punto a distancia cero de un punto dado es él mismo; ii) una distancia es una función simétrica; iii) una distancia satisface la desigualdad triangular: la longitud de uno de los lados de un triángulo es menor o igual a la suma de los otros dos lados. En la terminología del aprendizaje basado en distancias diremos que un elemento está “cerca” en lugar de decir que un elemento es “similar” a otro. Existen muchas distancias en la literatura para diversos tipos de datos. Las ecuaciones 1.1, 1.2, 1.3 muestran algunas de las distancias más conocidas en R^n .

Capítulo 1. Fundamentación teórica

$$\text{Distancia Euclídea } d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1.1)$$

$$\text{Distancia de Manhattan } d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (1.2)$$

$$\text{Distancia de Chebyshev } d(x, y) = \max_{1 < i < n} |x_i - y_i| \quad (1.3)$$

Generalización (11)

Intuitivamente, la generalización de un conjunto finito de elementos E en un espacio métrico (X, d) podría ser definida extensionalmente como un conjunto que contiene a E . Sin embargo, este tipo de definición por extensión no nos dice nada acerca del concepto o patrón que describe a todos los elementos de E . Por ejemplo, una generalización expresada como el conjunto de strings $ab, abab, ababab, abababab, \dots$ puede no resultar tan clara como la descripción o patrón $(ab)^+$.

En el ejemplo anterior podemos observar que hemos usado un lenguaje \mathcal{L} para dar las descripciones conceptuales. Asimismo, podemos notar que, dependiendo del lenguaje de patrones \mathcal{L} que elijamos, podremos expresar ciertas generalizaciones, pero no otras. Por ejemplo, el patrón que describe las cadenas de caracteres que contienen el mismo número de a 's que de b 's no puede ser expresado por el mismo lenguaje que usamos para el ejemplo anterior, es decir, el lenguaje de las expresiones regulares.

Diremos entonces que un patrón $p \in \mathcal{L}$ puede ser considerado una forma de representar los elementos de E y, al mismo tiempo, denotaremos con $\text{Set}(p)$ a los elementos descritos o cubiertos por p . Esto nos lleva a la noción de cobertura y de operador de generalización. Nos referiremos a $\text{Set}(p)$ como la cobertura del patrón p . Consecuentemente, diremos que un elemento $x \in X$ es cubierto por un patrón p si $x \in \text{Set}(p)$ y que p es una generalización de un conjunto E si y sólo si $E \subseteq \text{Set}(p)$. Para obtener una generalización o patrón expresado en un lenguaje de patrones \mathcal{L} , dada una evidencia E , necesitamos contar con una función que nos transforme los elementos de la evidencia en patrones que describan al menos esos elementos.

Definición 2 Sea (X, d) un espacio métrico, \mathcal{L} un lenguaje de patrones y $E \subseteq X$. Un operador de generalización es una función $\Delta_x: 2^X \rightarrow \mathcal{L}$ tal que $\Delta_x(E) = p$, donde $p \in \mathcal{L}$, $E \subseteq \text{Set}(p)$.

Capítulo 1. Fundamentación teórica

Por ejemplo, una posible generalización para un conjunto de puntos en \mathbb{R}^2 puede ser definida como el mínimo rectángulo que incluye todos los puntos en el conjunto (ver figura 3). En este caso, el lenguaje de patrones \mathcal{L} viene dado por el conjunto de los rectángulos.

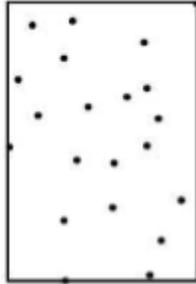


Figura 3: Una posible generalización de un conjunto de puntos en \mathbb{R}^2 .

Si restringimos E a conjuntos de dos elementos, denotamos con Δ al operador binario de generalización.

Definición 3 Sea (X,d) un espacio métrico y \mathcal{L} un lenguaje de patrones. Un operador binario de generalización es una función $\Delta: X \times X \rightarrow \mathcal{L}$ tal que dados $x_1, x_2 \in X$, $\Delta(x_1, x_2) = p$, donde $p \in \mathcal{L}$, $x_1, x_2 \in \text{Set}(p)$.

La figura 4 muestra cinco posibles generalizaciones de dos puntos en el espacio métrico (\mathbb{R}^2, d) , donde d es la distancia Euclídea.

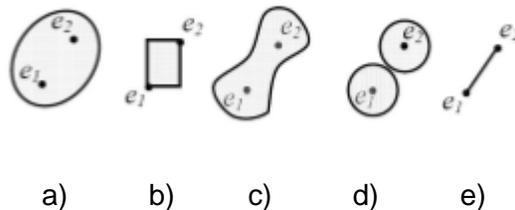


Figura 4: Cinco posibles generalizaciones de dos puntos en \mathbb{R}^2 .

De igual manera, así como generalizamos conjuntos de elementos o pares de elementos, podemos generalizar pares de patrones ya que los patrones describen elementos.

Definición 4 Sea (X,d) un espacio métrico y \mathcal{L} un lenguaje de patrones. Un operador binario de generalización de patrones es una función $\Delta^*: \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ tal que dados $p_1, p_2 \in \mathcal{L}$, $\Delta^*(p_1, p_2) = p$, donde $\text{Set}(p_i) \subseteq \text{Set}(p)$ ($i \in 1, 2$).

Capítulo 1. Fundamentación teórica

Se puede notar que para aquellos casos en los que el lenguaje de patrones \mathcal{L} es igual al conjunto X , los operadores Δ^* y Δ podrían ser definidos de igual manera.

En la figura 5 mostramos una posible generalización para dos patrones p_1 y p_2 en un lenguaje de patrones \mathcal{L} , donde \mathcal{L} es el conjunto de rectángulos de ejes paralelos. La generalización $\Delta^*(p_1, p_2)$ se encuentra definida como el mínimo rectángulo que cubre los patrones p_1 y p_2 .

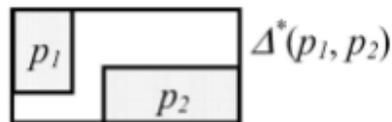


Figura 5: Una generalización de dos patrones en R^2 .

Agrupamiento jerárquico (11)

Los algoritmos de agrupamiento jerárquico construyen una jerarquía de grupos, llamada *dendrograma*, a partir de los elementos individuales que conforman la evidencia E . Las hojas de esta jerarquía corresponden a los elementos individuales, mientras que los nodos intermedios corresponden a subconjuntos de ejemplos que particionan la evidencia. En la figura 6 se muestra un ejemplo de un árbol de agrupamiento o dendrograma. El nodo raíz contiene todos los elementos de la evidencia, en este caso $E = \{A, B, C, D, E, F, G\}$. Cada nodo en el árbol se encuentra a un nivel diferente, el cual viene dado por la distancia existente entre los grupos. En general, para una evidencia formada por n ejemplos, el árbol tendrá n niveles, donde cada nivel i determina un agrupamiento diferente.

Capítulo 1. Fundamentación teórica

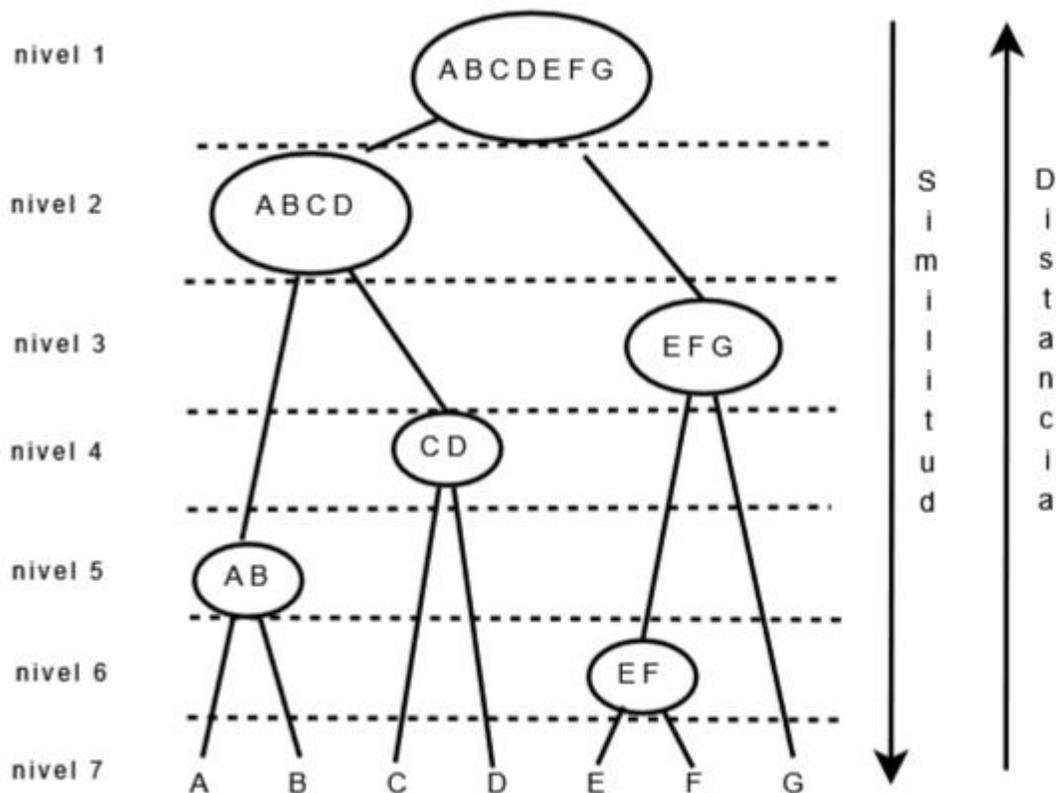


Figura 6: Un árbol de agrupamiento o dendrograma y sus n niveles.

Los grupos son unidos sobre la base de la distancia existente entre ellos, para lo cual se emplea una distancia entre grupos referida como *la distancia de enlace o enlazado*. Existen diversas distancias de enlace. Usualmente, la distancia de enlace entre dos grupos C_1 y C_2 viene determinada por la máxima distancia entre los elementos de cada grupo, y es conocida como distancia de enlace completa y que denotaremos a lo largo de este trabajo como d_L^c ; también puede estar dada por la mínima distancia entre los elementos de los grupos (distancia de enlace simple, d_L^s); por la distancia media entre los elementos de los grupos (distancia de enlace a la media, d_L^a) o también por la mínima distancia entre los prototipos de los grupos (distancia de enlace a los prototipos, d_L^p), entre otras.

Formalmente, dichas distancias se definen como sigue:

Sea (X,d) un espacio métrico, $C_1 \subseteq X$, $C_2 \subseteq X$ dos grupos de elementos de X ,

$$d_L^c(C_1, C_2, d) = \max \{d(x, y) : x \in C_1 \wedge y \in C_2\} \quad (1.4)$$

Capítulo 1. Fundamentación teórica

$$d_L^s(C_1, C_2, d) = \min \{d(x, y) : x \in C_1 \wedge y \in C_2\} \quad (1.5)$$

$$d_L^a(C_1, C_2, d) = \frac{1}{|C_1||C_2|} \sum_{x \in C_1} \sum_{y \in C_2} d(x, y) \quad (1.6)$$

$$d_L^p(C_1, C_2, d) = d(x_1, x_2) \quad (1.7)$$

Ecuación 1.4, 1.5, 1.6 y 1.7 para distancias entre grupos.

1.2 Herramientas informáticas afines

1.2.1 Herramientas internacionales

Sistema de RBC para el Soporte a la Toma de Decisiones

Los autores de este artículo muestran cómo el agente consejero del Sistema de Ayuda a la Elección de Asignaturas Optativas (SAEAO), empleando un sistema de RBC puede aconsejar a estudiantes de forma autónoma. Finalmente muestra los resultados obtenidos con él y los compara con los resultados arrojados por una red de funciones de base radial (RBF – Radial Basis Functions). Se presenta un método para la toma de decisiones sobre el problema expuesto anteriormente. La utilización del RBC muestra una mayor precisión a la hora de seleccionar el conjunto de asignaturas que la opción basada exclusivamente en la aplicación de una red neuronal artificial (12).

Sistema experto para la enseñanza: ENS-AI

ENS-AI, acrónimo de Enseñanza - Inteligencia Artificial es un sistema de soporte y guía para la práctica de la educación en España. Desarrollado en el Instituto de Investigación en Inteligencia Artificial del Consejo Superior de Investigaciones Científicas, utilizando el lenguaje Milord II. Es un sistema basado en reglas; esta facilidad permite representar pequeños dominios de conocimiento estructurados, como en el caso de la educación. El sistema posee las siguientes entradas de datos fundamentales, el alumno con sus características individuales y sociales, su entorno de enseñanza-aprendizaje con las características externas (entorno familiar y social) e internas (entorno escolar) y su profesor con sus características de formación, habilidades, destreza, actitudes y expectativas. Con todos estos datos realiza el diagnóstico psicopedagógico del alumno, así como una valoración de las técnicas y habilidades del profesor y del entorno de enseñanza-aprendizaje. Como resultado se obtiene la recomendación de una intervención

Capítulo 1. Fundamentación teórica

pedagógica (asociada a un valor de certeza), que le permite enfrentarse a la solución del problema diagnosticado (13).

Sistema experto de orientación vocacional – profesional

Sistema desarrollado en la Universidad Complutense de Madrid, España, que a partir del conocimiento de las aptitudes, personalidad e intereses del estudiante el sistema realiza un diagnóstico, como resultado aporta dos tipos de informaciones. En primer lugar le acercará a la profesión o profesiones (universitarias o no universitarias) que considera más adecuada en función de sus características personales. En segundo lugar le aporta conocimientos en relación a la profesión elegida (duración de los estudios, lugares donde puede estudiarse, salidas laborales, etc.). Es un sistema experto basado en reglas desarrollado en el lenguaje VP-Expert (14).

1.2.2 Herramientas nacionales

Sistema de RBC para la identificación de riesgos de software

Los autores de esta investigación, proponen una herramienta que utiliza el RBC para facilitar la identificación de riesgos en los proyectos de desarrollo de software. Asimismo, el sistema brinda la posibilidad de que a partir de un conjunto de características relevantes del proyecto que le son introducidas, se logre obtener de forma automática los riesgos potenciales a incidir a lo largo del ciclo de desarrollo, así como una posible mitigación o aprovechamiento de los mismos. Esto permite tener una visión adelantada, agilizando el proceso de toma de decisiones. La solución desarrollada combina principalmente un sistema de gestión de información con las técnicas de RBC. Constituye una aplicación web, basada en el modelo cliente - servidor y desarrollada sobre la plataforma Java (15).

El RBC en el ámbito de la Enseñanza/Aprendizaje

En este artículo, se describe un modelo para la enseñanza / aprendizaje, sustentado en la aplicabilidad del RBC para obtener una adecuada representación de la Modelación del Estudiante e inferir las estrategias de aprendizaje. Desarrollado en la Universidad Central de Las Villas, mediante el uso de la herramienta computacional HESSEI, la cual ha sido aplicada exitosamente en la toma de decisiones en tareas de enseñanza/ aprendizaje. Se utiliza como estructura de datos una BC donde los rasgos predictivos de los casos se corresponden con el módulo Conocimiento del Alumno, y el rasgo objetivo con el módulo

Capítulo 1. Fundamentación teórica

Conocimiento de los Contenidos, con el objetivo de identificar los propósitos que debe alcanzar un alumno al finalizar una unidad básica de aprendizaje (16).

Sistema Inteligente de Soporte a la Toma de Decisiones

En esta investigación, se propone un producto de software desarrollado completamente en software libre, que constituye una solución genérica, reutilizable y multiplataforma. Emplea técnicas de inteligencia artificial, específicamente el razonamiento basado en casos, permite a los dirigentes de cualquier institución u organismo tomar una decisión correcta en un tiempo mínimo, ante una situación dada. Con una arquitectura cliente servidor, este sistema, se consume desde la web como un servicio y puede ser perfectamente integrable tanto con un sistema de gestión como con un sistema de información geográfica que apoye también el proceso a informatizar (17).

Sistema experto para la definición de estrategias individuales de superación pedagógica

Una herramienta fiable para la toma de decisiones, durante el proceso de generación de estrategias individuales de superación pedagógica para la Universidad de las Ciencias Informáticas. Crea un mecanismo que reúne el conocimiento de los Jefes de Departamentos y de expertos en el tema de superación pedagógica; con el objetivo de: procesar tanta información de un profesor como se necesite, utilizar diversas vías de obtención del conocimiento sobre un profesor para el diseño de su superación pedagógica y aplicar el mecanismo de forma homogénea en todos los departamentos de la universidad (18).

El estudio comparativo de los sistemas descritos anteriormente y el propuesto en esta investigación, teniendo en cuenta una serie de parámetros, se muestran en la tabla 1. El estudio de las herramientas para el apoyo a la toma de decisiones permitió identificar un conjunto de funcionalidades básicas, así como componentes esenciales que debe poseer el sistema a desarrollar. Luego del análisis realizado se concluye que estas constituyen una base para el diseño e implementación de la propuesta, no para su utilización, ya que son aplicaciones específicas para una universidad o empresa determinada, que presentan características particulares en el dominio de aplicación. Incluso en (17), proponen un sistema con elementos significativos, no posee la opción de ser configurable, por tanto resulta limitado para la solución del problema de la investigación. Por otra parte, aunque

Capítulo 1. Fundamentación teórica

algunos resuelven problemáticas en el campo de la educación, pero no están enfocados en el proceso de elaboración de estrategias educativas.

Tabla 1: Tabla comparativa de las herramientas afines

Herramientas	Multiplataforma	Licencia Libre	Aplicado a la educación	Centrado en estudiantes	Enfocado al proceso de elaboración de estrategias educativas
1	X		X	X	
2			X	X	
3			X	X	
4	X				
5	X	X	X	X	
6	X	X			
7	X	X	X		
8	X	X	X	X	X

Leyenda:

- 1- Sistema de RBC para el Soporte a la Toma de Decisiones.
- 2- Sistema experto para la enseñanza: ENS-AI.
- 3- Sistema experto de orientación vocacional – profesional.
- 4- Sistema de RBC para la identificación de riesgos de software.
- 5- El RBC en el ámbito de la Enseñanza/Aprendizaje.
- 6- Sistema Inteligente de Soporte a la Toma de Decisiones.
- 7- Sistema experto para la definición de estrategias individuales de superación pedagógica.
- 8- Propuesta de los autores.

Capítulo 1. Fundamentación teórica

1.3 Metodología, lenguaje de programación y herramientas utilizadas

1.3.1 Herramientas

Actualmente se consideran a las Herramientas de Desarrollo de Software (HDS) como herramientas basadas en computadoras que asisten el proceso de ciclo de vida de una aplicación, consolidadas en la literatura en la forma de Ingeniería de software asistida por computadora (CASE, por sus siglas en inglés) (19).

Entorno de desarrollo integrado

Un IDE es un software compuesto por diferentes herramientas y funciones que permiten el trabajo con uno o más lenguajes de programación. Es un entorno que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Provee un marco de trabajo amigable para los lenguajes de programación. Entre los IDE que existen actualmente se encuentran: NetBeans, Eclipse y Visual Studio (19), para el desarrollo de la aplicación se escoge NetBeans en su versión 8.0.1.

NetBeans

NetBeans IDE es un entorno de desarrollo visual de código abierto para aplicaciones programadas mediante Java, uno de los lenguajes de programación más poderosos del momento. Su aprendizaje se ha convertido en fundamental para quienes están interesados en el desarrollo de aplicaciones multiplataforma. Mediante NetBeans es posible diseñar aplicaciones con solo arrastrar y soltar objetos sobre la interfaz de un formulario.

Con NetBeans IDE no solo es posible elaborar potentes aplicaciones para el Escritorio, también para la Web y para dispositivos portátiles, como móviles o Pocket PC, sin que cambie la forma de programar. La programación mediante NetBeans se realiza a través de componentes de software modulares, también llamados módulos. NetBeans pone a disposición de los usuarios decenas de módulos a través de su página web, que podrás integrar en él para conseguir mejores aplicaciones (20).

Gestor de Base de Datos

Es un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan

Capítulo 1. Fundamentación teórica

con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad (21).

PostgreSQL

PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python) (22), en nuestro trabajo se utilizó en su versión 9.4.1.

PgAdmin

PgAdmin es una aplicación gráfica de código abierto para gestionar las bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres (23).

PgAdmin está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad (23). En este trabajo se hace uso de PgAdmin en su versión 1.20.0 del 3 de Febrero de 2015.

Herramientas de modelado

Visual Paradigm for UML 8.0: Visual Paradigmes una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software

Capítulo 1. Fundamentación teórica

y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (24).

- Dibujo. Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta.
- Corrección sintáctica. Controla que el modelado con UML sea correcto.
- Coherencia entre diagramas. Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- Trabajo multiusuario. Permite el trabajo en grupo, proporcionando herramientas de compartición de trabajo.
- Generación de código. Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- Generación de informes. Permite generar diversos informes a partir de la información introducida en la herramienta.

Wolfram Mathematica

Mathematica es un sistema para aprender, hacer y aplicar la matemática en distintos campos del conocimiento. Su versatilidad para realizar cálculos en forma simbólica o numérica permite resolver problemas con cualquier grado de dificultad. El sistema maneja grandes volúmenes de datos con fines de análisis estadístico y ofrece al usuario capacidades inigualables para la visualización de resultados (25). En este trabajo se utiliza en su versión 10.

1.3.2 Lenguajes

Un lenguaje en el contexto informático es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por las máquinas, lo que proporciona un vocabulario para lograr una comunicación entre el equipo de desarrollo y los ordenadores (19).

Lenguaje de Modelado

Lenguaje Unificado de Modelado (UML 2.0): “UML es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. Fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para

Capítulo 1. Fundamentación teórica

modelar aplicaciones de muy diversos dominios de aplicación y plataformas de objetos distribuidos” (26).

- Permite construir todos los artefactos necesarios para llevar a cabo un proceso de desarrollo con calidad (requisitos, arquitectura, pruebas y versiones).
- Modela estructuras complejas.
- Desde su aparición es ampliamente utilizado en la industria del software.
- Es un lenguaje muy expresivo, que permite cubrir todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Permite especificar todas las decisiones de análisis, diseño e implementación; construyéndose así modelos más precisos y completos.

Se selecciona UML 2.0 ya que mejora sustancialmente el desarrollo de los proyectos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

Para modelar usando el lenguaje UML existe gran variedad de herramientas CASE, que permiten generar los artefactos que corresponden a cada una de las etapas del desarrollo. Una de esas herramientas es Visual Paradigm for UML 8.0.

Lenguaje de Programación Java

Es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de Lenguaje de Programación C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria (27).

La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos (27).

A diferencia de C++, Java no dispone de operadores de sobrecarga definidos por el usuario. Los diseñadores de Java tomaron esta decisión puesto que consideraban que, bajo ciertas circunstancias, esta característica podía complicar la lectura y mantenimiento

Capítulo 1. Fundamentación teórica

de los programas (27). En este trabajo se utilizará este lenguaje en su versión Java SE Development Kit 8 Update 66.

Lenguaje de consulta estructurado SQL

SQL es un lenguaje estandarizado que sirve para definir y manipular los datos de una base de datos relacional. De acuerdo con el modelo relacional de datos, la base de datos se crea como un conjunto de tablas, las relaciones se representan mediante valores en las tablas y los datos se recuperan especificando una tabla de resultados que puede derivarse de una o más tablas base. Las sentencias de SQL las ejecuta un gestor de bases de datos. Todas las sentencias de SQL ejecutables deben prepararse antes de su ejecución. El resultado de esta preparación es el formato operativo o ejecutable de la sentencia. El método de preparación de una sentencia de SQL y la persistencia de su formato operativo diferencian SQL estático de SQL dinámico (28).

SQL está basado en el cálculo relacional de tuplas. Como resultado, toda consulta formulada utilizando el cálculo relacional de tuplas (o su equivalente, el álgebra relacional) se puede formular también utilizando SQL. Hay, sin embargo, capacidades que van más allá del cálculo o del álgebra relacional (22).

1.3.3 Framework de desarrollo

Un framework de desarrollo es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones (29). Para el desarrollo de la aplicación se escoge Swing y se utiliza la librería Swing Layouts Extensions 1.0.4.

Swing

El paquete Swing es un paquete gráfico para Java que está compuesto por un amplio conjunto de componentes de interfaces de usuario que funcionan en el mayor número posible de plataformas. Además ofrece un conjunto de componentes escritos en Java con

Capítulo 1. Fundamentación teórica

más y mejores funcionalidades y con la independencia de plataforma que propone la tecnología Java.

Es un framework MVC para desarrollar interfaces gráficas para Java con independencia de la plataforma. Sigue un simple modelo de programación por hilos, y posee las siguientes características principales:

- Independencia de plataforma.
- Extensibilidad: Es una arquitectura altamente particionada, los usuarios pueden proveer sus propias implementaciones modificadas para sobrescribir las implementaciones por defecto. Se puede extender clases existentes proveyendo alternativas de implementación para elementos esenciales.
- Personalizable: Los usuarios pueden proveer su propia implementación de apariencia, que permitirá cambios uniformes en la apariencia existente en las aplicaciones Swing sin efectuar ningún cambio al código de aplicación (30).

1.3.4 Metodología XP

Metodología ágil basada en cuatro principios: simplicidad, comunicación, retroalimentación y valor. Además, orientada por pruebas y refactorización, se diseña e implementan las pruebas antes de programar la funcionalidad, el programador crea sus propios tests de unidad (31).

La metodología Xp está enfocada a grupos pequeños y medianos de construcción de software en donde los requisitos aún son muy ambiguos, cambian rápidamente o son de alto riesgo. Xp busca la satisfacción del cliente tratando de mantener durante todo el tiempo su confianza en el producto (31).

Actividades de Xp

Escuchar: "Los programadores no lo conocemos todo, y sobre todo muchas cosas que las personas de negocios piensan que son interesantes. Si ellos pudieran programarse su propio software ¿para qué nos querrían?".

Si se va a hacer pruebas se tiene que preguntar si lo obtenido es lo deseado, y hay que preguntar quién necesita la información. Hay que escuchar a los clientes, cuáles son los problemas de su negocio, hay que tener una escucha activa, explicando lo que es fácil y

Capítulo 1. Fundamentación teórica

difícil de obtener, y la realimentación entre ambos ayudan a todos a entender los problemas.

Diseñar: El diseño crea una estructura que organiza la lógica del sistema. Un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.

Codificar: Es necesario codificar y plasmar nuestras ideas mediante el código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar.

Hacer pruebas: Las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que el trabajo funciona, cuando no se puede pensar en ninguna prueba que pudiese originar un fallo en el sistema, entonces se habrá acabado por completo (31).

Se selecciona la metodología XP para el desarrollo del presente trabajo debido a que es un proyecto pequeño y de corto tiempo, donde no se necesita la generación de tantos artefactos y los requisitos pueden cambiar con el tiempo a medida que avanza el trabajo; además, el equipo de desarrollo está compuesto por dos programadores, en el cual el grado de interacción entre los miembros es alto, pues existe una buena comunicación y entendimiento.

Conclusiones

En este capítulo se analizaron los principales conceptos a emplear para el desarrollo de la investigación de las dos esferas que interactúan en esta investigación: la pedagógica y la informática. En el área educativa se adoptaron posiciones pedagógicas sobre lo que es y lo que implica la elaboración de una estrategia educativa, sobre la definición y características del diagnóstico, una variable que se concreta en dimensiones y éstas a su vez en un conjunto de indicadores, que representan una muestra de los aspectos a tener en cuenta en la elaboración de la estrategia, en un proceso como el cubano que tenga en su centro la instrucción, la formación y el desarrollo del educando.

Capítulo 1. Fundamentación teórica

En el área informática se realizó un estudio del estado del arte para un mejor entendimiento de la herramienta que se quiere desarrollar. Se hizo un análisis detallado de la metodología, herramientas y lenguajes a utilizar para el correcto desarrollo de la solución, definiendo como metodología guía la XP, como entorno de desarrollo el Eclipse en su versión del 2013, Juno Service Release 2, hasta una versión avanzada de la aplicación, y para culminarla se hace uso de NetBeans 8.0.1. Como herramienta de modelado Visual Paradigm en su versión 8.0 y como SGBD se seleccionó PostgreSQL en su versión 9.4.1. Como lenguaje de programación Java en su versión Java SE Development Kit 8 Update 66 y como framework de desarrollo Swing.

Capítulo 2. Propuesta de solución

Capítulo 2. Propuesta de solución

Introducción

En el presente capítulo se realizará un análisis de las principales características del sistema a desarrollar. Se realiza un análisis de los requisitos funcionales que deben estar presentes en el sistema. Se exponen los artefactos generados propios de la metodología de desarrollo utilizada para la implementación de la solución. Se realiza una descripción de las historias de usuario que se proponen para dar cumplimiento al objetivo general originado por la situación problemática. Se realiza el plan de duración de las iteraciones a llevar a cabo.

2.1 Propuesta de solución

Tras analizar la situación actual en la que se encuentra la Universidad de Ciencias Informáticas, debido a la gran cantidad de estudiantes que conviven en ella, a la diversidad de las características de los mismos y al gran volumen de información que se archiva sobre un estudiante en particular. Surge así la necesidad de desarrollar una herramienta informática que permita agrupar a estos estudiantes con características comunes, lo que permitiría seleccionar eficientemente las acciones que se podrían llevar a cabo, para una adecuada elaboración de las estrategias educativas por los docentes y poder dar una atención diferenciada, tanto grupal como individual, atendiendo así a uno de los principios de la Didáctica de la atención a las diferencias individuales.

Como propuesta de solución al problema de esta investigación, se plantea que dicha herramienta informática servirá como apoyo al proceso de elaboración y puesta en práctica de las estrategias educativas en la Facultad 2 de la UCI. Esta herramienta tomará de la base de datos la información almacenada de cada estudiante (objeto), donde cada característica (rasgo) será tomada como una dimensión del espacio. Si se tienen n características el espacio fuese de R^n , quedando estos representados como puntos en el mismo. Para determinar cuán similares son dos objetos se utilizó una fórmula de distancia, la cual expresa que a menor distancia mayor similitud y al ser cero existe una igualdad entre los objetos. Para apoyar el cálculo de distancia se empleó un umbral de distancia, el cual determina la máxima distancia aceptable para clasificar a dos objetos como semejantes.

Capítulo 2. Propuesta de solución

Esta herramienta permite agrupar los elementos, dígame puntos o grupos, mediante una generalización, que se basa en tomar la mínima área que encierre los elementos. Para caracterizar dicha área se crea un patrón que abarque el rango de características de todos los que se encuentran dentro de la misma.

Al mismo tiempo que se agrupan elementos, las comparaciones ya no serían entre objetos solamente, sino entre cualquier combinación de objetos y grupos, por tanto surge la necesidad de poder calcular la distancia cuando interviene un grupo. Para esto se utilizó la distancia de enlace que nos permite resolver esta situación.

Teniendo en cuenta lo anteriormente planteado se agruparán los elementos que se encuentren a la distancia más pequeña, los más similares, creando el patrón respectivo a ese agrupamiento. Este proceso se repetirá hasta que no hallan elementos similares, por tanto que se encuentren a una distancia mayor que el umbral. Una explicación más detallada de lo descrito anteriormente se puede ver a continuación:

2.1.1 Declaración de la base de casos

Los rasgos de los estudiantes están dados por los indicadores de las dimensiones de la variable diagnóstico, los cuales se identificaron mediante la documentación analizada y entrevistas a expertos y a profesores de experiencia, mostrándose en la operacionalización de la variable diagnóstico.

La operacionalización de la variable diagnóstico precisa un grupo de dimensiones, con sus respectivos indicadores e ítems. Si bien en la operacionalización no ha de incluirse el tipo de dato (tema tratado por informáticos), se cree conveniente incluirlo para facilitar la lectura del documento. La tabla que a continuación se presenta contiene los utilizados en la investigación.

Tabla 2: Operacionalización de la variable diagnóstico

ORDEN DE PRIORIDAD	DIMENSIÓN	INDICADORES (TIPO DE DATO)	ITEMS
1	Motivos e	Gusto por la carrera (Nu)	Escala [1-5]

Capítulo 2. Propuesta de solución

(instructivas)	intereses	Rama que más le gusta (No)	*(abierto)
		Interés cognoscitivos(Nu)	Escala [1-5]
1 (instructivas)	Académico	Rendimiento escolar (Nu)	Escala [1-5]
		Tipo de aprendizaje (O)	-rápido -medio -lento
		¿Posee un método de estudio efectivo? (B)	[si,no]
		Nivel de desarrollo de habilidades intelectuales (Nu)	Escala [1-5]
		Nivel de desarrollo de habilidades docentes (Nu)	Escala [1-5]
		Nivel de desarrollo de habilidades procedimentales (Nu)	Escala [1-5]
		Participación en las actividades docentes (Nu)	Escala [1-5]
		Participación en las actividades extradocentes (Nu)	Escala [1-5]
		Uso de las TIC (Nu)	Escala [1-5]
		Nivel de actualización de conocimientos técnicos (Nu)	Escala [1-5]

Capítulo 2. Propuesta de solución

		Tipo de centro de procedencia (No)	-preuniversitario -tecnológico -camilitos o pre militar -facultad -traslado de carrera
		Vía de ingreso a la Universidad (No)	-cadetes MINFAR -cadetes MININT -concurso -orden 18 -preuniversitario
2 (formativos)	Social	Nivel de actualización nacional e internacional (O)	-alto -medio -bajo
		Nivel de comunicación (O)	-alto -medio -bajo
		Participación en actividades culturales (Nu)	Escala [1-5]
		Participación en actividades deportivas (Nu)	Escala [1-5]
		Participación en actividades artísticas (Nu)	Escala [1-5]

Capítulo 2. Propuesta de solución

		Participación en actividades políticas (Nu)	Escala [1-5]
		Integración al grupo (Nu)	Escala [1-5]
		Colaboración y apoyo a sus compañeros (Nu)	Escala [1-5]
		Sensibilidad por los problemas del grupo e individuales (Nu)	Escala [1-5]
2 (formativos)	Valores	Honestidad (Nu)	Escala [1-3]
		Responsabilidad (Nu)	Escala [1-3]
		Compromiso (Nu)	Escala [1-3]
		Sentido de pertenencia a su país (Nu)	Escala [1-5]
		Conocimiento de los símbolos patrios (Nu)	Escala [1-5]
		Conocimiento de la Historia del país (Nu)	Escala [1-5]
2 (formativos)	Normas de comportamiento	Respeto a sus compañeros y profesores (Nu)	Escala [1-5]
		Trato cordial y afable (Nu)	Escala [1-5]
		Cumplimiento de las normas establecidas (Nu)	Escala [1-5]
		Asistencia a clases (Nu)	Escala [1-5]
3	Nivel de vida	Refrigerador (B)	[si, no]

Capítulo 2. Propuesta de solución

(influye en el desarrollo personal)		Cocina de gas o eléctrica (B)	[si, no]
		Radio (B)	[si, no]
		TV (B)	[si, no]
		Celular (B)	[si, no]
		Laptop (B)	[si, no]
		Computadora (B)	[si, no]
		Tablet (B)	[si, no]
		Aire acondicionado (B)	[si, no]
		Automóvil (B)	[si, no]
3 (influye en el desarrollo personal)	Relación familiar	Nivel cultural de los que conviven (O)	-universitario -preuniversitario -técnico medio -noveno grado -menor de noveno
		Padres divorciados (B)	[si, no]
		Número de personas del núcleo familiar (Nu)	*(abierto)
4	Datos generales	Sexo (B)	[Masculino, Femenino]
		Raza (No)	[Blanco, Negro, Mestizo]
		Edad (Nu)	*(abierto)

Capítulo 2. Propuesta de solución

		Provincia de residencia (No)	-Pinar del Rio -La Habana -Artemisa -Mayabeque -Isla de la Juventud -Matanzas -Villa Clara -Cienfuegos -Sancti Spiritus -Las Tunas -Camagüey -Holguín -Granma -Santiago de Cuba -Guantánamo
--	--	---------------------------------	---

* Los Indicadores con ítems de carácter abierto significa que los posibles valores se irán creando a medida que los estudiantes vayan dando sus respuestas.

Leyenda:

Nu: Numérico

No: Nominal

O: Ordinal

B: Binario

Objeto: estudiante

Rasgos, su tipo y dominio

Nombre: Este rasgo no se tiene en cuenta dentro de la operacionalización de la variable ni en el proceso de agrupamiento, por tanto no se establece un dominio pero sirve para identificar a los estudiantes.

Los rasgos están representados en los indicadores de las dimensiones de la operacionalización de la variable diagnóstico, el tipo se muestra entre paréntesis junto al indicador, ejemplo rasgo (tipo), y el dominio de cada rasgo se evidencia en la columna "ítems", todo esto se muestra en la tabla 2.

2.1.2 Funciones de comparación de rasgos

Función de comparación para datos binarios y nominales:

Los datos nominales se van a comportar como los binarios.

Capítulo 2. Propuesta de solución

$$\delta(x, y) = \begin{cases} 1, & \text{si } x \neq y \\ 0, & \text{si } x = y \end{cases} \quad (2.1)$$

Donde x y y son rasgos de e_i y e_j respectivamente; $e_i, e_j \in E$.

Función de comparación para datos numéricos y ordinales:

$$\delta(x, y) = \frac{|x - y|}{\Delta S} \quad (2.2)$$

Donde x y y son rasgos de e_i y e_j respectivamente; $e_i, e_j \in E \wedge S$ el dominio del rasgo.

2.1.3 Función de distancia o semejanza (d)

Se utilizará la ecuación de distancia Euclídea (ver ecuación 1.1).

Esta función unida a la evidencia E son usadas para hallar la matriz de semejanza.

2.1.4 Umbral de distancia o semejanza

El umbral es la cota inferior de los posibles valores de semejanza existentes en el momento de comparar dos objetos (e_i y e_j). Su función consiste en restringir el intervalo en caso de que se desee lograr una mayor precisión en la obtención de los casos más semejantes.

La magnitud $d_0 \in \Delta$ ($\Delta = [0, 1]$, $\Delta = \{0, 1\}$, $\Delta = \{1, \dots, k\}$, y otros) se denominará umbral de semejanza y puede ser calculada, por ejemplo, con las siguientes ecuaciones 2.3, 2.4 y 2.5 (6):

$$d_0 = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n d(I(O_i), d(O_j)) \quad (2.3)$$

$$d_0 = \max_{\substack{i=1, \dots, n-1 \\ i \neq j}} \left\{ \min_{j=i+1, \dots, n} \{d(I(O_i), d(O_j))\} \right\} \quad (2.4)$$

$$d_0 = \min_{\substack{i=1, \dots, n-1 \\ i \neq j}} \left\{ \max_{j=i+1, \dots, n} \{d(I(O_i), d(O_j))\} \right\} \quad (2.5)$$

Capítulo 2. Propuesta de solución

Se utilizará en este trabajo la ecuación 2.3 puesto que tiene una visión un tanto más general de las distancias entre evidencias debido a que trabaja en base al promedio.

2.1.5 Generalización

Debido a la demostrada importancia que tiene la selección correcta de la generalización a utilizar, se escogió una ya antes probada como eficaz y eficiente (11), además mencionada en el capítulo 1, esta puede ser definida en \mathbb{R}^2 como el mínimo rectángulo que incluye todos los puntos en el conjunto. Para poder utilizarla como operador binario en la creación de nuevos patrones se debe tener en cuenta si es punto a punto (Figura 4.b)) o entre dos patrones ya creados (Figura 5).

A la generalización seleccionada se le realizó una pequeña adaptación porque el caso analizado en este trabajo tiene n dimensiones y esta generalización es solo para dos. Esto no afecta en nada a la generalización en sí, por lo que podemos tomar la lógica de cómo trabajar con las dos dimensiones para aplicarla a nuestro problema n dimensional.

2.1.6 Distancia de enlace

Para el cálculo de la distancia entre grupos se decidió aplicar la distancia de enlace a la media (ver Ecuación 1.6) por tener en cuenta todas las distancias entre los elementos de los grupos en comparación y no solo la máxima o la mínima, las cuales pueden ser más apropiada para otros tipos de problemas que solo requieran de ese criterio a tener en cuenta en el momento de referirse a la distancia entre grupos. Otro criterio por el que se hizo la selección es para ser consecuente con la ecuación de umbral de distancia seleccionada y poder descartar semejanza entre grupos que tengan elementos que no cumplan con la condición impuesta por el umbral.

2.1.7 Algoritmo HDCC (Modificado)

Entrada:

$E = \{e_1, e_2, \dots, e_n\} \subseteq X$, una distancia d , con (X, d) un espacio métrico;

Δ^* : $\mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$ un operador binario de generalización de patrones;

Δ : $X \times X \rightarrow \mathcal{L}$ un operador binario de generalización;

d_L : $2^X \times 2^X \times (X \times X \rightarrow \mathbb{R}) \rightarrow \mathbb{R}$ una distancia de enlace.

Capítulo 2. Propuesta de solución

Salida:

Una lista T de grupos y generalizaciones.

Pasos:

1. Insertar $(\{e_i\}, \Delta(e_i, e_i), \{0\})$ como elemento de T, para todo $e_i \in E$.
2. $MS \leftarrow$ Calcular $d(e_i, e_j)$ entre cada par $e_i, e_j \in E$ con $i < j$.
3. Calcular $d_0(MS)$.
4. $MS \leftarrow \infty$ para todo $(MS_i, MS_j) > d_0$ con $i < j$.
5. Insertar $d_L(C_i, C_j, MS)$ entre cada par de grupos $C_i, C_j \in T$ con $i < j$, usando la distancia MS.
6. Buscar distancia mínima global.
7. Mientras (distancia mínima global $> d_0$)
 - 7.1. Calcular el patrón pc_{xy} del grupo C_{xy} como $\Delta^*(pc_x, pc_y)$, donde $C_{xy} = C_x \cup C_y$; pc_x, pc_y son los patrones de C_x y C_y respectivamente, y C_x y C_y son los grupos más cercanos en T de acuerdo a d_L .
 - 7.2. Calcular C_{xyz} con $C_{xyz} = C_{xy} \cup C_z$ y $C_z = \{e \mid e \in C_i \wedge C_i \in T \wedge C_i \subseteq \text{Set}(pc_{xy})\}$.
 - 7.3. Insertar (C_{xyz}, pc_{xy}) en T como primer elemento.
 - 7.4. $T \leftarrow T - \{C_i\}$ para todo C_i tal que $C_i \subseteq \text{Set}(pc_{xy})$.
 - 7.5. Eliminar las $d_L(C_i, C_j)$, para todo C_i, C_j tal que $C_i \subseteq \text{Set}(pc_{xy})$ o $C_j \subseteq \text{Set}(pc_{xy})$.
 - 7.6. Calcular $d_L(C_{xyz}, C_i, MS)$ para todo C_i tal que $C_i \in T$ con $i > 1$.
 - 7.7. Buscar distancia mínima global.
8. Retornar T.

2.2 Funcionalidades del sistema

Los requerimientos funcionales son declaraciones de los servicios que proveerá el sistema, de la manera en que éste reaccionará a entradas particulares, deben ser lo más completo, claro y preciso posible. En algunos casos, los requerimientos funcionales de los sistemas también declaran explícitamente lo que el sistema no debe hacer (32).

A continuación se detallan las funcionalidades del sistema. El sistema debe ser capaz de:

- Gestionar estudiantes.
- Agrupar estudiantes.

Capítulo 2. Propuesta de solución

- Mostrar patrón por agrupamientos.
- Mostrar estudiantes por agrupamientos.
- Importar datos de estudiantes desde fichero .csv.

2.3 Propiedades del producto

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y la representación de datos que se utiliza en la interface del sistema.

Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas de software o hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad, entre otros (32).

2.3.1 Usabilidad

- El sistema podrá ser usado por cualquier persona que posee elementos básicos de computación.
- Los elementos que conforman una interfaz de usuario deben tener un nombre sugerente, que logre que el usuario perciba la acción.
- El sistema debe contar con un estilo de presentación coherente para cada ventana.

2.3.2 Interfaz

- El sistema debe tener una interfaz fácil, amigable y sencilla para que los usuarios finales del mismo puedan interactuar con este aun teniendo conocimientos básicos.

2.3.3 Rendimiento

- Los tiempos de respuestas del sistema deben ser rápidos, al igual que la velocidad de procesamiento de la información para lograr respuestas rápidas del mismo.

Capítulo 2. Propuesta de solución

2.3.4 Restricciones de diseño

- El producto de software final debe diseñarse sobre una arquitectura cliente-servidor.

2.4 Fase de Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (33).

2.4.1 Historias de usuarios

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. El cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Las historias de usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración (33).

Tabla 3: Historia de usuario # 1: Gestionar estudiantes

Historia de Usuario	
Número: 1	Usuario: Usuario
Nombre de historia: Gestionar estudiantes.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 1

Capítulo 2. Propuesta de solución

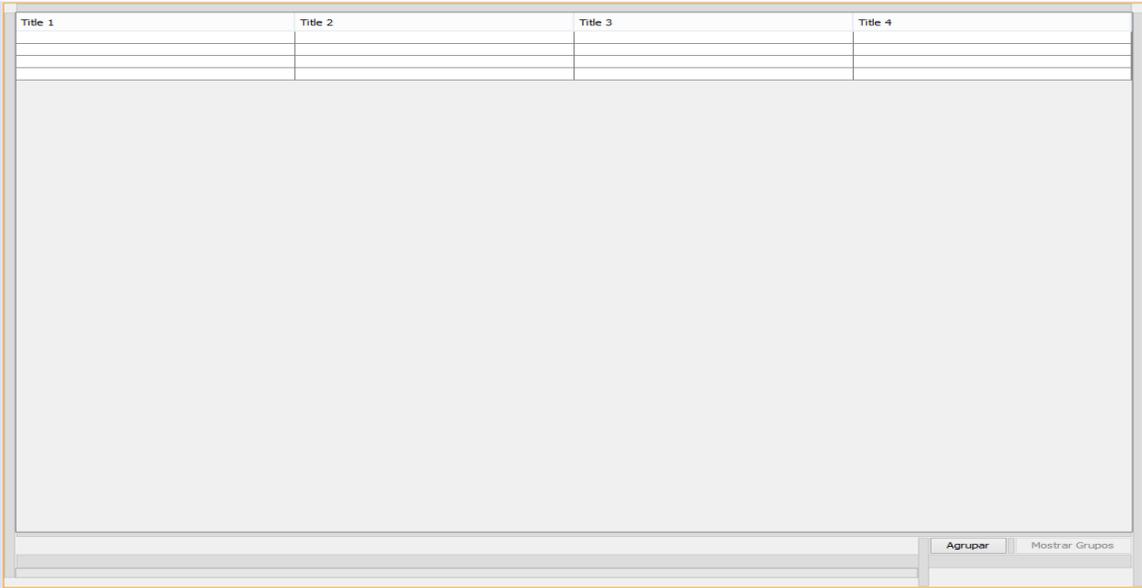
Programadores responsables: Jorge Raúl Gómez, Anniel Sánchez
Descripción: El sistema debe permitir mostrar todos los estudiantes que se encuentran almacenados en la base de datos. Así como insertar uno nuevo y modificar o eliminar los existentes.
Observaciones:
Prototipo de Interfaz:


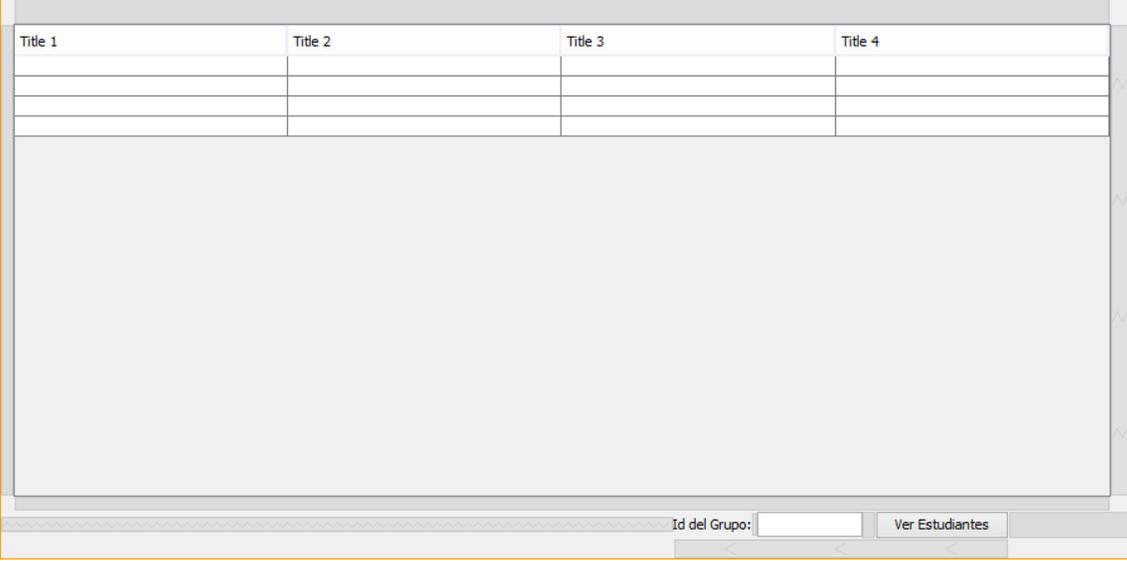
Tabla 4: Historia de usuario # 2: Agrupar estudiantes

Historia de Usuario	
Número: 2	Usuario: Usuario
Nombre de historia: Agrupar estudiantes.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 1
Programadores responsables: Jorge Raúl Gómez, Anniel Sánchez	
Descripción: El sistema debe permitirle al usuario, a través de una opción, agrupar a	

Capítulo 2. Propuesta de solución

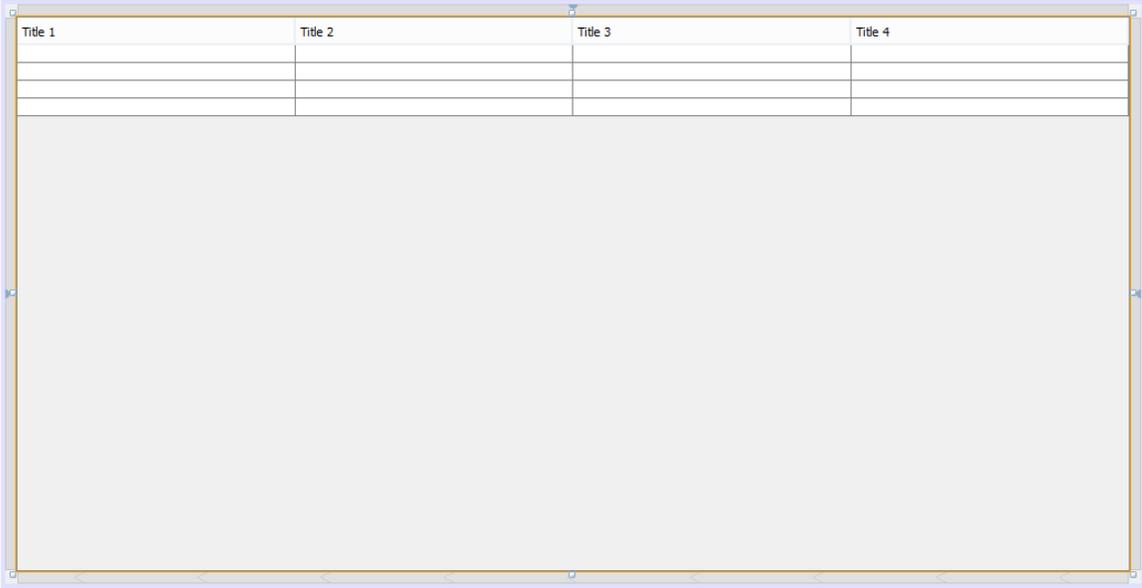
todos los estudiantes.
Observaciones:
Prototipo de Interfaz: No aplica

Tabla 5: Historia de usuario # 3: Mostrar patrón por agrupamiento

Historia de Usuario	
Número: 3	Usuario: Usuario
Nombre de historia: Mostrar patrón por agrupamiento.	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 2
Programadores responsables: Jorge Raúl Gómez, Anniel Sánchez	
Descripción: El sistema permite mostrar el patrón perteneciente a cada agrupamiento de estudiantes.	
Observaciones:	
Prototipo de Interfaz:	
	

Capítulo 2. Propuesta de solución

Tabla 6: Historia de usuario # 4: Mostrar estudiantes por agrupamiento

Historia de Usuario	
Número: 4	Usuario: Usuario
Nombre de historia: Mostrar estudiantes por agrupamiento.	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 2
Programadores responsables: Jorge Raúl Gómez, Anniel Sánchez	
Descripción: Permite mostrar todos los estudiantes de los diferentes agrupamientos que han sido formados anteriormente a partir de la preferencia del usuario.	
Observaciones:	
Prototipo de Interfaz:	
	

Capítulo 2. Propuesta de solución

Tabla 7: Historia de Usuario # 5: Importar datos de estudiantes desde Excel

Historia de Usuario	
Número: 5	Usuario: Usuario
Nombre de historia: Importar datos de estudiantes desde fichero .csv.	
Prioridad en negocio: Media	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 3
Programadores responsables: Jorge Raúl Gómez, Anniel Sánchez	
Descripción: El sistema debe permitirle al usuario, importar datos desde archivos de Excel que contengan los mismos campos que se miden en la herramienta.	
Observaciones:	
Prototipo de Interfaz: No aplica	

2.5 Fase de Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración (33).

2.5.1 Estimación de esfuerzos por Historias de Usuario

Para el desarrollo satisfactorio de la solución propuesta, se realizó una estimación de esfuerzo para cada una de las historias de usuario, arrojando los siguientes resultados:

Capítulo 2. Propuesta de solución

Tabla 8: Estimación de esfuerzo por Historia de Usuario

Número	Historia de Usuario	Puntos estimados
1	Gestionar estudiantes.	3 semanas
2	Agrupar estudiantes.	2semana
3	Mostrar patrón por agrupamiento.	2 semana
4	Mostrar estudiantes por agrupamiento.	1 semana
5	Importar datos de estudiantes desde fichero.csv.	2 semanas

2.5.2 Plan de Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción (33).

Tomando como referencia los aspectos antes tratados la aplicación que se pretende construir se desarrollará en 3 iteraciones, explicadas más detalladamente a continuación:

Iteración 1

La primera iteración del desarrollo del sistema, tendrá como objetivo la implementación de las funcionalidades básicas para la puesta en marcha del mismo. Se tendrán en cuenta las historias de usuario con prioridad de negocio alta, o sea se llevará a cabo la implementación de las Historias de Usuario 1 y 2. Al terminar la iteración esto representará un 50% de la solución.

Capítulo 2. Propuesta de solución

Tabla 9: Plan de duración de la iteración 1

Iteraciones	Historias de Usuario	Duración
Iteración 1	Gestionar estudiantes	5 semanas
	Agrupar estudiantes	

Iteración 2

Durante la segunda iteración se seguirá con la implementación de historias de usuario de prioridad media, en este caso las Historias de Usuario número 3 y 4. Al terminar la iteración esto representará un 80% de la solución.

Tabla 10: Plan de duración de la iteración 2

Iteraciones	Historias de Usuario	Duración
Iteración 2	Mostrar patrón por agrupamientos	3 semanas
	Mostrar estudiantes por agrupamientos	

Iteración 3

Durante la tercera iteración se desarrollará la historia de usuario de prioridad media con las cual se completarán todas las funcionalidades para las que está diseñado el sistema. Se llevará a cabo la implementación de la Historia de Usuario número 5. Al terminar la iteración esto representará un 100% de la solución.

Tabla 11: Plan de duración de la iteración 3

Iteraciones	Historias de Usuario	Duración
Iteración 3	Importar datos de estudiantes desde fichero .csv	2 semana

Capítulo 2. Propuesta de solución

Conclusiones

En este capítulo se describió la propuesta de solución a desarrollar. Se definieron las funcionalidades a implementar y las propiedades con las que debe cumplir el producto, así como la descripción de las historias de usuario divididas por tres iteraciones y la estimación del esfuerzo dedicado a la realización de cada una de ellas en el orden en que se les dará cumplimiento.

Capítulo 3. Implementación y Prueba

Capítulo 3. Diseño, implementación y prueba

Introducción

En este capítulo se describen las fases de implementación y prueba, propias de la metodología XP. Se define la arquitectura a utilizar en la solución. Se adapta el diseño al entorno de implementación donde se define el patrón arquitectónico que se utilizará, se realiza un análisis de los patrones de diseño que se utilizará para su correcta implementación y se definen las tarjetas Clase-Responsabilidad-Colaborador (CRC) para identificar y organizar las clases orientadas a objetos. Se expone el modelo físico de la base de datos. Se describen las tareas de ingeniería generadas para cada historia de usuario. Por último, se realizan las pruebas al producto para comprobar su correcto funcionamiento y si las funcionalidades se han implementado de forma adecuada.

3.1 Arquitectura de software

“La IEEE Std 1471-2000 define la Arquitectura del Software como la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”, o sea es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación (34).

3.1.1 Arquitectura Cliente-Servidor

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda del cliente que la produjo. Mediante esta arquitectura el usuario (cliente) puede acceder a la información sin tener en cuenta su ubicación física y donde pueda estar alojada la misma. A continuación se muestra en la figura 7 la arquitectura cliente-servidor.

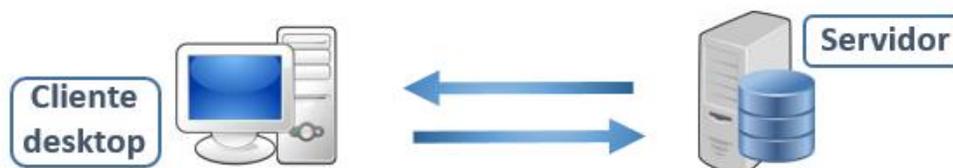


Figura 7: Arquitectura Cliente-Servidor

Capítulo 3. Implementación y Prueba

3.2 Fase de diseño

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

Simplicidad: Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Soluciones “spike”: Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “spike”), para explorar diferentes soluciones. Estos programas son únicamente para probar o evaluar una solución, y suelen ser desechados luego de su evaluación.

Recodificación: La recodificación (“refactoring”) consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible. Muchas veces, al terminar de escribir un código de programa, pensamos que, si lo comenzáramos de nuevo, lo hubiéramos hecho en forma diferente, más clara y eficientemente (35).

A la hora de darle cumplimiento a la actividad de diseñar, XP no especifica ninguna técnica de modelado. Pueden utilizarse indistintamente sencillos esquemas en una pizarra, diagramas de clases utilizando UML o tarjetas CRC (Clase, Responsabilidad y Colaboración) siempre que sean útiles, tributen a la comprensión y no requieran mucho tiempo en su creación.

3.2.1 Patrón Arquitectónico

Un patrón arquitectónico describe una categoría del sistema que contiene: un conjunto de componentes (por ejemplo, una base de datos, módulos computacionales) que realizan una función requerida por el sistema; un conjunto de conectores que posibilitan la comunicación, la coordinación y la cooperación entre los componentes; restricciones que definen cómo se pueden integrar los componentes que forman el sistema; y modelos semánticos que permiten al diseñador entender las propiedades globales de un sistema para analizar las propiedades conocidas de sus partes constituyentes (36). Para el

Capítulo 3. Implementación y Prueba

desarrollo de este trabajo el patrón arquitectónico a seguir es el Modelo–Vista–Controlador (MVC).

Modelo–Vista–Controlador

El framework que se utilizó para el desarrollo de la solución es Swing, el cual usa el Modelo–Vista–Controlador (MVC). El patrón MVC es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo (37).

Definición de las partes (37)

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

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

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

A continuación la figura 8 muestra el funcionamiento del patrón MVC:

Capítulo 3. Implementación y Prueba

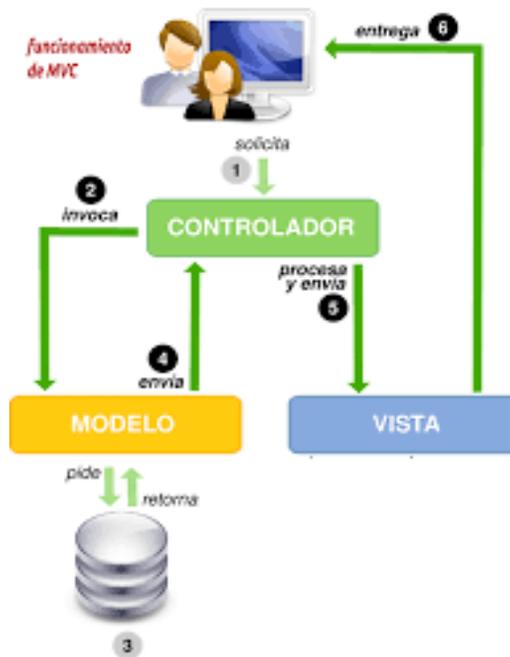


Figura 8: Patrón Modelo -Vista -Controlador

Este patrón comienza su funcionamiento cuando el usuario envía una solicitud al controlador, este invoca al modelo que se encarga de interactuar con la base de datos para obtener el resultado de la solicitud y a su vez le devuelve la información al controlador que procesa dicha información y mediante una vista se la entrega al usuario.

3.2.2 Patrones de diseño

Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (38).

Patrón GRASP

Los patrones GRASP son parejas de problema solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades (39).

Patrón Experto

Problema: ¿Cuál es el principio fundamental en virtud del cual se asignan las responsabilidades en el diseño orientado a objetos?

Capítulo 3. Implementación y Prueba

Solución: Asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad (39). Este patrón se verá reflejado en la clase Principal.

Patrón Creador

Problema: ¿Quién debería ser responsable de crear una nueva instancia de alguna clase?

Solución: La creación de objetos es una de las actividades más frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella (39). Este patrón se verá reflejado en las clases: Grupo y Patrón.

Bajo Acoplamiento

Problema: ¿Cómo dar soporte a una dependencia escasa y a un aumento de la reutilización?

Solución: Asignar una responsabilidad para mantener bajo acoplamiento (39). Este patrón se verá reflejado en las clases Patrón y Grupo.

Alta Cohesión

Problema: ¿Cómo mantener la complejidad dentro de límites manejables?

Solución: Asignar una responsabilidad de modo que la cohesión siga siendo alta (39). Este patrón se verá reflejado en las clases Patrón, Grupo y Principal.

Patrón Controlador

Problema: ¿Quién debería encargarse de atender un evento del sistema?

Solución: Asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones:

- el sistema global (controlador de fachada).
- la empresa u organización global (controlador de fachada).
- algo en el mundo real que es activo (por ejemplo, el papel de una persona) y que pueda participar en la tarea (controlador de tareas) (39).

Este patrón se verá reflejado en la clase Principal.

Capítulo 3. Implementación y Prueba

3.2.3 Tarjeta Clase –Responsabilidad –Colaborador (CRC)

Las tarjetas CRC son una metodología para el diseño de programas orientados a objetos, resumen las responsabilidades de una clase antes de pasar a realizar el diagrama y sirven para simular escenarios. Las tarjetas CRC representan objetos; la clase a la que pertenece el objeto se puede escribir arriba, las responsabilidades u objetivos que debe llevar a cabo el objeto se pueden escribir a la izquierda, y a la derecha las clases que se relacionan con dicho objeto (40).

La metodología de desarrollo XP como parte de la fase de diseño propone el modelado de Clase-Responsabilidad-Colaborador (CRC), lo que constituye un modelo simple de organizar las clases para las funcionalidades del sistema, con el objetivo de desarrollar una representación organizada de las clases. A continuación se muestran las tarjetas CRC.

Tabla 12: Tarjeta CRC: Patrón

Clase: Patrón	
Responsabilidad	Colaborador
Contener la estructura de un patrón. Obtener los atributos de dicha estructura. Modificar los atributos de dicha estructura.	Ninguna

Tabla 13: Tarjeta CRC: Grupo

Clase: Grupo	
Responsabilidad	Colaborador
Contener la estructura de un grupo. Obtener los atributos de dicha estructura. Modificar los atributos de dicha estructura. Crear un nuevo patrón a través de la unión de dos ya existentes.	Patrón

Capítulo 3. Implementación y Prueba

Tabla 14: Tarjeta CRC: Principal

Clase: Principal	
Responsabilidad	Colaborador
<p>Conectar con la base de datos.</p> <p>Obtener los datos de la tabla estudiantes.</p> <p>Insertar los patrones finales en su tabla correspondiente.</p> <p>Insertar la pertenencia de los estudiantes a cada patrón.</p> <p>Calcular distancia entre todos los estudiantes.</p> <p>Calcular distancia entre grupos de estudiantes.</p> <p>Hallar la distancia mínima entre estudiantes o grupos (elemento) y determinar entre quienes se establecen.</p> <p>Crear un nuevo grupo con los elementos de la distancia mínima hallada.</p> <p>Identificar los elementos que no fueron captados por la distancia mínima pero que si cumplen con el nuevo patrón creado e insertarlos en el nuevo grupo.</p> <p>Eliminar los elementos con los que se ha creado el nuevo grupo.</p>	<p>Patrón, Grupo</p>

Tabla 15: Tarjeta CRC: Visual

Clase: Visual	
Responsabilidad	Colaborador
<p>Mostrar todos los estudiantes que están almacenados.</p> <p>Permite agrupar los estudiantes.</p> <p>Mostrar el tiempo de ejecución del algoritmo que se encarga de agrupar.</p> <p>Crea la interfaz gráfica de la clase Visual2.</p>	<p>Principal, Visual2</p>

Capítulo 3. Implementación y Prueba

Tabla 16: Tarjeta CRC: Visual2

Clase: Visual2	
Responsabilidad	Colaborador
Mostrar los patrones de los agrupamientos creados. Validar la información entrada por el usuario en el cuadro de texto respectivo al id del grupo a mostrar. Crear la interfaz gráfica de la clase Visual3.	Visual, Visual3

Tabla 17: Tarjeta CRC: Visual3

Clase: Visual3	
Responsabilidad	Colaborador
Mostrar los estudiantes pertenecientes al grupo seleccionado por el usuario.	Visual2

3.3 Modelo físico de la base de datos

Un modelo de datos es una colección de herramientas conceptuales para describir los datos, las relaciones de datos, la semántica de los datos y las ligaduras de consistencia (41).

A continuación en la figura 9 se muestra el modelo físico de la base de datos, este modelo muestra la relación que existe entre las entidades de las bases de datos.

Capítulo 3. Implementación y Prueba

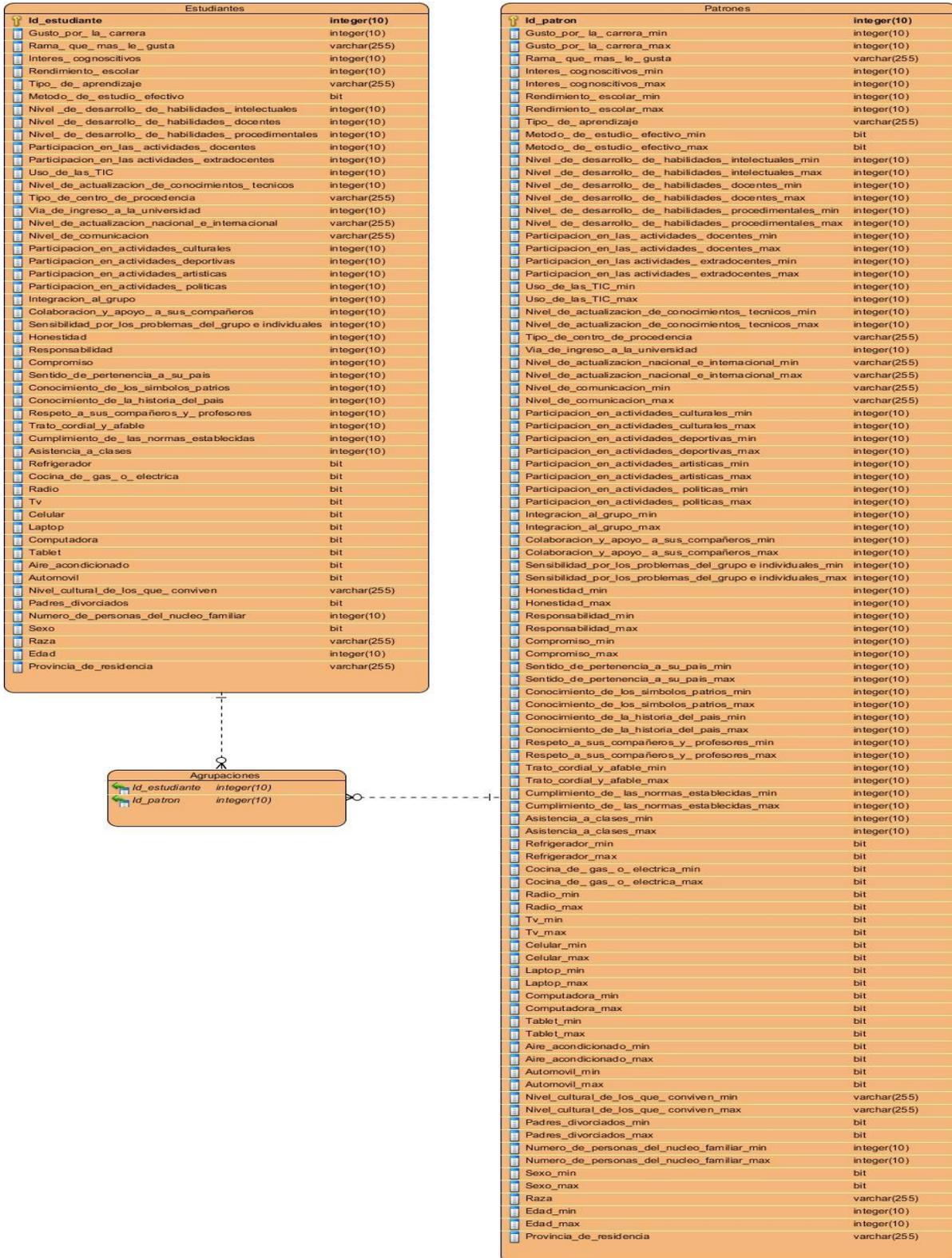


Figura 9: Modelo físico de la base de datos

Capítulo 3. Implementación y Prueba

3.4 Implementación

Como se ha detallado anteriormente un proyecto desarrollado en XP se divide en iteraciones para realizar la codificación, de manera que se puede entregar una versión funcional del software, que aunque no esté completo en su totalidad, irá creciendo ciclo a ciclo. En cada iteración, se implementarán las historias de usuario definidas previamente en el Plan de Iteraciones. Se define además un conjunto de tareas, que deberán realizar los programadores para llevar a cabo la construcción del sistema.

Implementación de la 1ra Iteración

Durante la primera iteración de la construcción del sistema se implementaron las historias de usuario que son de prioridad alta para poner en marcha el sistema, donde se obtuvo la primera versión del mismo. En esta iteración se le dio cumplimiento a las tareas de ingeniería correspondientes a las historias de usuario de la primera iteración.

Implementación de la 2da Iteración

Durante la implementación de la segunda iteración del software se perfeccionaron las funcionalidades de la primera entrega y se continuó con el desarrollo de la tarea correspondiente a la historia de usuario con prioridad media que se desarrolla en esta iteración.

Implementación de la 3ra Iteración

Una vez concluida la implementación de las historias de usuario de la tercera iteración de la construcción del software, se obtuvo la versión completa del producto. Se realizó la tarea correspondiente a la historia de usuario de la tercera iteración.

3.5 Tareas de ingeniería

Las tareas de ingeniería son escritas por los desarrolladores, a partir de las historias de usuario creadas por el cliente, las cuales brindan una mayor información para la implementación. Las tareas de ingeniería se representan mediante tablas, a continuación se muestra una tabla donde se explican las secciones:

Capítulo 3. Implementación y Prueba

Tabla 18: Ejemplo de una Tarea de Ingeniería

Tarea de Ingeniería	
Número tarea: (Número de la tarea, debe ser consecutivo)	Número historia de usuario: (Número de la historia de usuario a la que pertenece la tarea)
Nombre tarea: (Nombre que identifica la tarea)	
Tipo de tarea: (Las tareas se pueden clasificar en: desarrollo, mejora, corrección u otra)	Puntos estimados: (Tiempo que se le estimará al desarrollo de la tarea)
Fecha inicio: (Fecha en que inicia el desarrollo de la tarea)	Fecha fin: (Fecha en que culmina el desarrollo de la tarea)
Programador Responsable: (Nombre y apellidos del programador responsable)	
Descripción: (Breve descripción de la tarea)	

A continuación se muestran las tareas de ingeniería:

Tabla 19: Tarea de Ingeniería # 1: Gestionar estudiantes

Tarea de Ingeniería	
Número tarea: 1	Número historia de usuario: 1
Nombre tarea: Gestionar estudiantes.	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 5 -2 - 2016	Fecha fin: 26 -2 - 2016
Programador Responsable: Jorge Raúl Gómez, Anniel Sánchez	
Descripción: Al ejecutar la herramienta, esta automáticamente genera una tabla con contenido informacional de los estudiantes. Consta de cinco botones; uno de ellos con label 'Agrupar' se encarga de generar los grupos y sus patrones respectivos, el segundo con label 'Mostrar Grupos' se encarga de crear la interfaz gráfica de la clase Visual2. El tercero, cuarto y quinto se encargan de insertar, modificar y eliminar respectivamente los estudiantes.	

Tabla 20: Tarea de Ingeniería # 2: Agrupar estudiantes

Tarea de Ingeniería

Capítulo 3. Implementación y Prueba

Número tarea: 2	Número historia de usuario: 2
Nombre tarea: Agrupar estudiantes.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 1 -3 - 2016	Fecha fin: 15 -3 - 2016
Programador Responsable: Jorge Raúl Gómez, Aniel Sánchez	
Descripción: Se encarga de implementar el algoritmo HDCC (Modificado).	

Tabla 21: Tarea de Ingeniería # 3: Mostrar patrón por agrupamiento

Tarea de Ingeniería	
Número tarea: 3	Número historia de usuario: 3
Nombre tarea: Mostrar patrón por agrupamiento.	
Tipo de tarea: Mejora	Puntos estimados: 2
Fecha inicio: 23 -4 - 2016	Fecha fin: 7 -5 - 2016
Programador Responsable: Jorge Raúl Gómez, Aniel Sánchez	
Descripción: Al generarse esta ventana, se muestra una tabla con los patrones correspondientes a los grupos creados. Consta de un cuadro de texto, en el cual se puede escribir el id de un grupo específico para luego ser mostrados los estudiantes pertenecientes al mismo mediante el botón, con label 'Ver Estudiantes'.	

Tabla 22: Tarea de Ingeniería # 4: Mostrar estudiantes por agrupamiento

Tarea de Ingeniería	
Número tarea: 4	Número historia de usuario: 4
Nombre tarea: Mostrar estudiantes por agrupamiento.	
Tipo de tarea: Mejora	Puntos estimados: 1
Fecha inicio: 22 -5 - 2016	Fecha fin: 29 -5 - 2016
Programador Responsable: Jorge Raúl Gómez, Aniel Sánchez	
Descripción: Al generarse esta ventana, se muestra una tabla con los estudiantes correspondientes al grupo seleccionado por el usuario.	

Capítulo 3. Implementación y Prueba

Tabla 23: Tarea de ingeniería # 5: Importar datos de estudiantes desde fichero .csv

Tarea de Ingeniería	
Número tarea: 5	Número historia de usuario: 5
Nombre tarea: Importar datos de estudiantes desde fichero .csv.	
Tipo de tarea: Mejora	Puntos estimados:2
Fecha inicio: 8 -6 - 2016	Fecha fin: 22 -6 - 2016
Programador Responsable: Jorge Raúl Gómez, Anniel Sánchez	
Descripción: Permite, dada la ubicación de un fichero .csv, cargar los datos del mismo e incorporarlos a las base de datos.	

3.6 Pruebas

Unos de los pilares de la metodología XP es el proceso de pruebas. XP propone comprobar constantemente cada una de las funcionalidades implementadas. Esto permite aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. La metodología XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación o funcionales.

3.6.1 Pruebas de Caja Blanca

En estas pruebas estamos siempre observando el código, que las pruebas se dedican a ejecutar con ánimo de "probarlo todo". Esta noción de prueba total se formaliza en lo que se llama "cobertura" y no es sino una medida porcentual de cuánto código hemos cubierto (42).

Prueba del camino básico

El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la determinación de un conjunto básico de caminos de ejecución. Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia de programa (43).

Capítulo 3. Implementación y Prueba

Complejidad Ciclomática

Es una medida que proporciona una idea de la complejidad lógica de un programa. Su cálculo se expresa como se muestra en la ecuación 3.1:

$$V(G) = \text{Aristas} - \text{Nodos} + 2 \quad (3.1)$$

A partir del valor de la complejidad ciclomática obtenemos el número de caminos independientes, los cuales son todas las posibles opciones por las que puede transitar el algoritmo.

En esta investigación se le realizó la prueba del camino básico a fragmentos de código seleccionados de la herramienta informática, tales como el cálculo de la distancia de enlace y el umbral de semejanza.

Camino básico para la distancia de enlace:

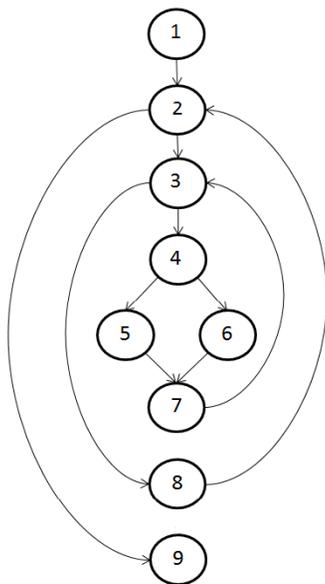


Figura 10: Prueba de camino básico para el cálculo de la distancia de enlace

Complejidad ciclomática para el cálculo de la distancia de enlace:

Nodos = 9

Aristas= 11

Complejidad Ciclomática = $11 - 9 + 2 = 4$.

Capítulo 3. Implementación y Prueba

Caminos Independientes:

1-2-3-4-5-7-3-8-2-9

1-2-3-4-6-7-3-8-2-9

1-2-3-8-2-9

1-2-9

Camino básico para el umbral de semejanza:

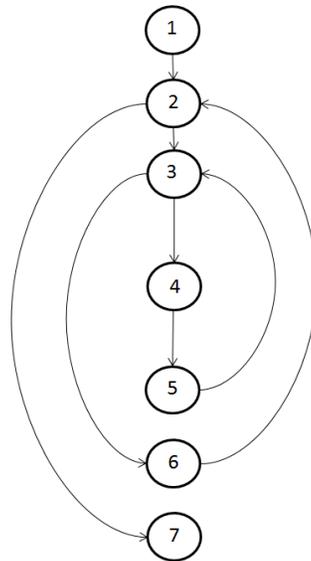


Figura 11: Prueba de camino básico para el cálculo del umbral de semejanza

Complejidad Ciclomática para el cálculo del umbral de semejanza:

Nodos = 7

Aristas= 8

Complejidad Ciclomática = $8 - 7 + 2 = 3$.

Caminos Independientes:

1-2-3-4-5-3-6-2-7

1-2-3-6-2-7

1-2-7

Capítulo 3. Implementación y Prueba

3.6.2 Prueba de Rendimiento

El servicio de pruebas de rendimiento de software se centra en determinar la velocidad con la que el sistema bajo pruebas realiza una tarea en las condiciones particulares del escenario de pruebas (44).

Con el objetivo de estudiar el rendimiento de la herramienta representamos en la figura 12 la dependencia del tiempo necesario para la ejecución de nuestro programa con el número de estudiantes analizados. Para ello se ha planteado, para un valor fijo del eje de las abscisas, la media aritmética de estos tiempos cada 10 corridas del programa. Se observa un comportamiento no lineal: la dependencia corresponde a una ley de potencia, que se aprecia también en la figura 12 para un exponente igual a 2.

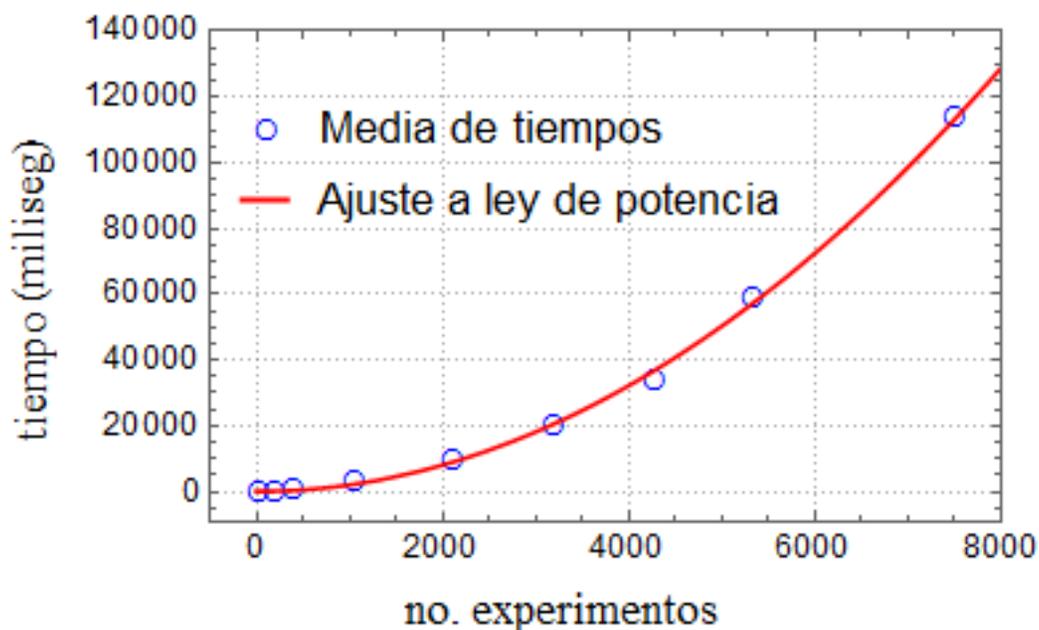


Figura 12: Dependencia de la media de los tiempos de corrida contra cantidad de estudiantes almacenados base de datos

Antes de seguir con el análisis estadístico que impera luego de esta observación, cabe aclarar lo siguiente: ¿qué rol desempeñan las fluctuaciones correspondientes a un valor dado del número de estudiantes?, ¿por qué no implementar un gráfico de barras de error?

Para ello puede observarse en la figura 13 cómo decrece $\Delta n/\bar{n}$ con el aumento de los valores de la variable independiente, donde \bar{n} y Δn son la media y desviación estándar para cada experimento respectivamente, -llamamos experimento a los conjuntos de 10 corridas correspondientes a cada conglomerado de estudiantes -. Nótese que la media de

Capítulo 3. Implementación y Prueba

las ordenadas corresponde a un valor de 0.04, es decir, que como promedio se observan desviaciones que son aproximadamente solo un 4% de la media de los tiempos de corrida. En otras palabras, no es necesario un análisis con barras de error; las fluctuaciones no son significativas.

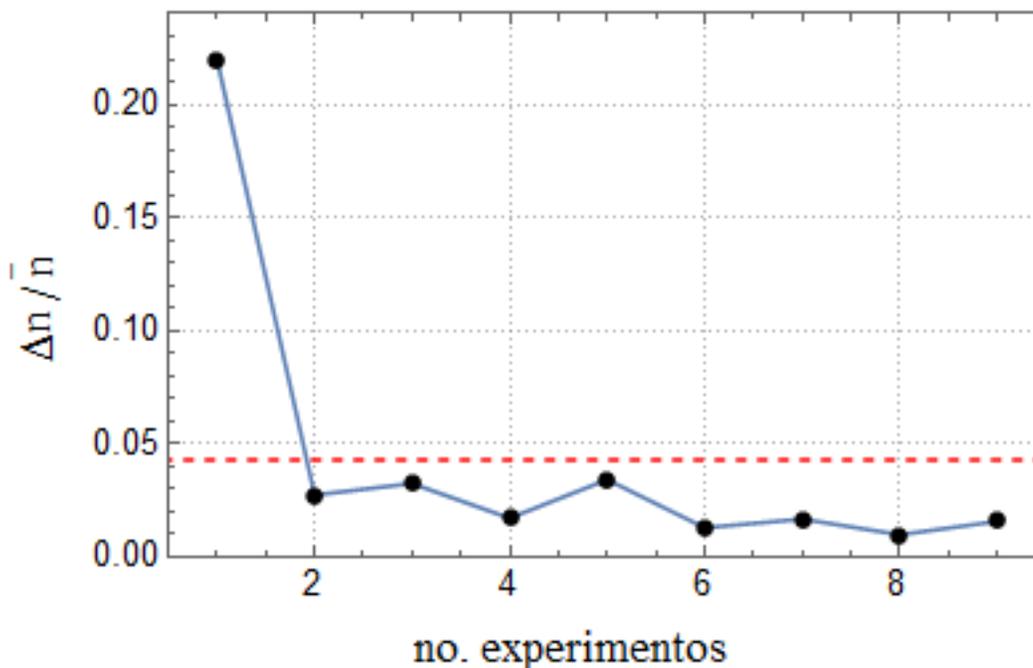


Figura 13: Decrecimiento de $\Delta n / \bar{n}$ con el aumento del número de experimentos

Dicho esto se prosigue, con el fin de caracterizar el ajuste elegido, haciendo un gráfico Log-Log para linealizar y encontrar el correspondiente coeficiente de regresión lineal (R^2), estudiando así la bondad del ajuste. La figura 14 muestra la data y la curva de ajuste para un $R^2=0.97$, lo que habla de la buena elección de la forma funcional de la curva analítica. Se obtiene entonces, para concluir la caracterización del rendimiento del programa, que el ajuste presentado en la figura 12 sigue la ley: $y = \alpha x^2$ donde $\alpha = (2.01 \pm 0.02) \times 10^{-3}$.

Capítulo 3. Implementación y Prueba

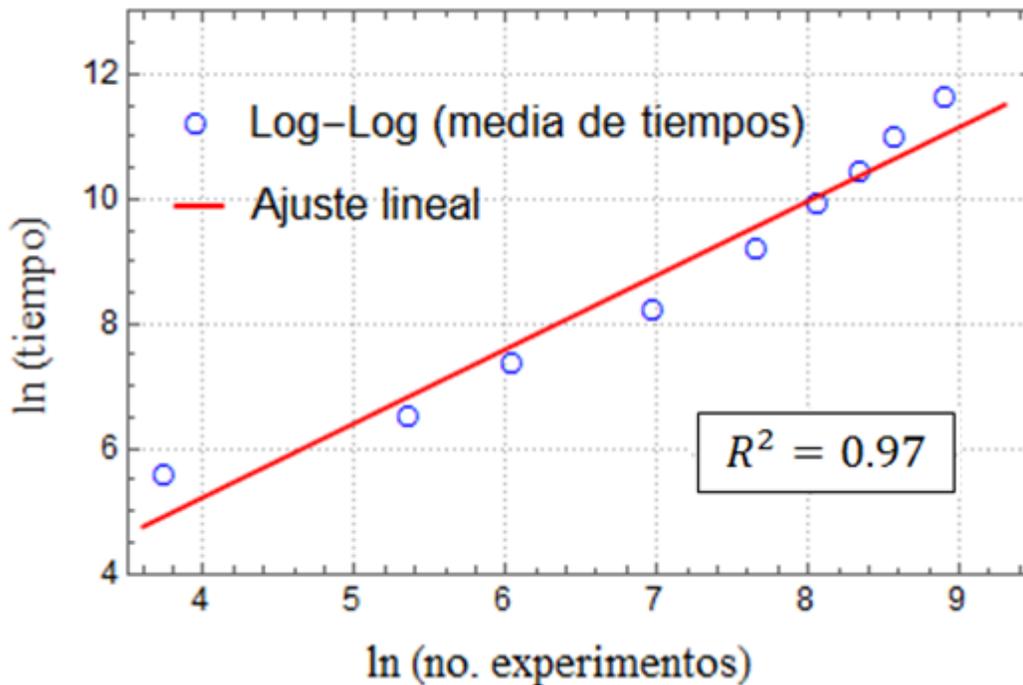


Figura 14: Gráfico Log-Log para la media de los tiempos de ejecución que muestra un coeficiente de regresión lineal de 0.97

Conclusiones

En este capítulo se estudió y se definió la arquitectura cliente servidor, para así diseñar y guiar la realización de la Herramienta informática para el apoyo al proceso de elaboración de las estrategias educativas. Se realizó un análisis para seleccionar el patrón arquitectónico a utilizar y posteriormente se seleccionaron los patrones de diseño que se utilizarán para el desarrollo del sistema. Se describieron las tareas de Ingeniería para cada historia de usuario, necesarias para la implementación del sistema. Además se definieron las tarjetas CRC permitiendo identificar y organizar las clases orientadas a objetos. Se expuso el modelo físico de la base de datos. Se llevaron a cabo las pruebas de caja blanca, permitiendo comprobar el correcto funcionamiento de procedimientos internos del software. Con la realización de la prueba de rendimiento se logró medir cuán rápido funciona la herramienta en dependencia de los datos analizados.

Conclusiones generales

Conclusiones Generales

Una vez finalizada la investigación se arribaron a las conclusiones siguientes:

- ✓ Se determinaron desde concepciones pedagógicas e informáticas los elementos consustanciales de la herramienta informática, donde se utilizan los datos que ofrece el diagnóstico para contribuir al mejoramiento del proceso de elaboración de las estrategias educativas.
- ✓ Se aplicaron los conceptos fundamentales, relacionados con las estrategias educativas y el importante rol del diagnóstico educativo en ellas, así como los requisitos de los algoritmos de agrupamiento conceptual, que posibilitaron el cumplimiento del objetivo previsto de la investigación.
- ✓ Teniendo en cuenta los requisitos de los algoritmos de agrupamiento conceptual se seleccionó el algoritmo utilizado en la herramienta informática.
- ✓ Se realizó un estudio de las herramientas informáticas existentes actualmente para el agrupamiento de objetos, tanto internacionales como nacionales, especialmente las de la UCI, determinando las características comunes de las mismas y evidenciando la necesidad de una, capaz de resolver la problemática existente en la Facultad 2 de la Universidad de Ciencias Informáticas.
- ✓ Se precisaron las tecnologías y herramientas, así como la metodología utilizada en el desarrollo de la investigación, lo que posibilitó implementar exitosamente la propuesta de solución.
- ✓ A partir de las concepciones pedagógicas e informáticas, así como de la precisión de los requisitos de la agrupación conceptual se elaboró la propuesta de solución.
- ✓ La herramienta informática elaborada permite el agrupamiento de los estudiantes según características similares para el apoyo al proceso de elaboración de las estrategias educativas en la Facultad 2 de la Universidad de Ciencias Informáticas.
- ✓ Se realizaron pruebas de validación que permitieron la identificación y corrección de no conformidades existentes en la aplicación, garantizando la obtención de una versión estable de la herramienta informática.

Conclusiones generales

Por todo lo anteriormente expuesto, se concluye que los objetivos propuestos para el presente trabajo se han cumplido satisfactoriamente, poniendo en práctica todas y cada una de las tareas propuestas para el desarrollo de la herramienta informática para la agrupación de estudiantes.

Recomendaciones

Recomendaciones

Teniendo en cuenta los resultados obtenidos en la realización del presente trabajo de diploma se recomienda:

- ✓ Ampliar el diagnóstico aplicado a los estudiantes, en donde se puedan agregar nuevas características y/o profundizar en algunas de las existentes, que se determine que sean de importancia para el proceso de elaboración de las estrategias educativas.
- ✓ Continuar el desarrollo de la investigación con el objetivo de perfeccionar y aumentar las funcionalidades de la aplicación.
- ✓ Extender el uso de la herramienta informática, no solo a la Facultad 2, sino a toda la Universidad de las Ciencias Informáticas.

Referencias

1. Fiallo, Jorge P., Cerezal , Julio y Hedesa, Ysidro J. *La investigación pedagógica una vía para elevar la calidad educativa*. Lima : s.n., 2008.
2. Cantoni, Nélica M. Técnicas de muestreo y determinación del tamaño de la muestra en investigación cuantitativa. *Revista Argentina de Humanidades y Ciencias Sociales*.
3. Calero, Julio y Lorente, María del Carmen. *La aplicación del juicio de experto como técnica de evaluación de las tecnologías de la información y comunicación (TIC)*. Sevilla : s.n.
4. de Armas, Nerelys y Valle, Alberto. *Resultados Científicos en la Investigación Educativa*. 2011.
5. Silvestre, Margarita y Zilberstein, José. *¿Cómo hacer más eficiente el aprendizaje?* Lima : s.n., 2001.
6. *Acerca del surgimiento del Reconocimiento de Patrones en Cuba* . Shulcloper, José Ruiz. 2, La habana : s.n., 27/5/2013, Vol. 7. 2227-1899.
7. Ruiz-Shulcloper , José. *RECONOCIMIENTO LÓGICO COMBINATORIO DE PATRONES: TEORÍA Y APLICACIONES* . 2009.
8. Silveira Martineaux , Karina y Fernández Pérez, Reidelendy. *Comparación de algoritmos de clasificación y agrupamiento aplicando técnicas de Minería de datos*. 2008.
9. Diaz, Darel Camps. *Implementación de algoritmos* . 2007.
10. Fernandes, Juan Vivanco. *Trabajo de T.I.C II*.
11. Funes, Ana. *Agrupamiento Conceptual Jerárquico Basado en Distancias* . España : s.n., 12/2008.
12. Laza, R, Fernández, F y Corchado. *Sistema de RBC para el Soporte a la Toma de Decisiones*. 2012.
13. Barraso, Clara. ENS-AI: UN SISTEMA EXPERTO PARA LA ENSEÑANZA. *Teoría de la Educación*. 1994, Vol. 6.

Referencias

14. Martín, María José. *SISTEMA EXPERTO DE ORIENTACIÓN VOCACIONAL PROFESIONAL*. 1996.
15. Cordero, Dasiel, Ruiz, Yadira y Torres, Yoanny. Sistema de RBC para la identificación de riesgos de software. *Revista Cubana de Ciencias Informáticas*. 2013, Vol. 7.
16. Martínez, Natalia. *El RBC en el ámbito de la Enseñanza/Aprendizaje*. 2010.
17. Rodríguez, Kathrin. *Sistema Inteligente de Soporte a la Toma de Decisiones*. 2010.
18. García, Harold y Alamino, Yuniel. *SISTEMA EXPERTO PARA LA GENERACIÓN DE ESTRATEGIAS INDIVIDUALES DE SUPERACIÓN PEDAGÓGICA DEL CLAUSTRO DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*. La Habana : s.n., 2014.
19. Ramón Galán , Yadira y Bécquer del Toro , Yordany. *Sistema para la evaluación de algoritmos de agrupamiento para el reconocimiento de patrones*. 2013.
20. NetBeans IDE. [En línea] [Citado el: 5 de 6 de 2016.] netbeans-ide.uptodown.com/windows.
21. Saz, Jesús Tramullas. *Introducción a la documática*. 2010.
22. Lockhart, Thomas. *El equipo de desarrollo de PostgreSQL*. 1996.
23. pgadmin. [En línea] [Citado el: 3 de 5 de 2016.] <http://www.pgadmin.org/>.
24. *Guión Visual Paradigm for UML*. 2013-2014.
25. Universidad Nacional Autónoma de México. [En línea] [Citado el: 5 de 6 de 2016.] <http://www.software.unam.mxpreguntas.html>.
26. Fuentes , Lidia y Vallecillo, Antonio . *Una Introducción a los Perfiles UML*.
27. Ecured. [En línea] 2005. [Citado el: 30 de 3 de 2016.] http://www.ecured.cu/index.php/Lenguaje_de_programaci%C3%B3n_Java?PageSpeed=noscript.
28. IBM Knowledge Center. [En línea] [Citado el: 20 de 4 de 2016.] http://www.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.sql.ref.doc/doc/c0004100.html?lang=es.
29. Gutiérrez, Javier J. *¿Qué es un framework web?*

Referencias

30. Blog de Jorgenio. [En línea] 2012. [Citado el: 30 de 5 de 2016.] <https://blog.jorgenio.com/category/swing/>.
31. Villegas, Adrian Anaya. A propósito de programación extrema XP (eXtreme Programming). [En línea] [Citado el: 28 de 4 de 2016.] <http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema.shtml#ixzz474f3NfGi>.
32. *Metodología Gestión de Requerimientos*. [En línea] sites.google.com.
33. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Letelier, Patricio. 26, Buenos Aires : s.n., 2006, Vol. 5. 1666-1680.
34. Casanova, Josep. DesarrolloWeb.com. [En línea] 9 de 10 de 2004. [Citado el: 10 de 5 de 2016.] <http://www.desarrolloweb.com/articulos/1622.php>.
35. Yolanda, Borja López. *Metodología Ágil de Desarrollo de Software - XP*.
36. Pressman, Roger S. *Ingeniería del software*. 2002.
37. *Patrón Modelo-Vista-Controlador*. Fernández Romero, Yenisleidy y Díaz González, Yanette. 1, 2012, Vol. 11.
38. Paredes Niz, Arleth. Ingeniería de software. [En línea] 27 de 8 de 2012. [Citado el: 30 de 5 de 2016.] <https://arlethparedes.wordpress.com/2012/08/27/patrones-de-arquitectura-vs-patrones-de-diseno>.
39. Visconti, Marcello y Astudillo, Hernán. *Fundamentos de Ingeniería de Software*.
40. Programación Orientada a Objetos y Taller. [En línea] 29 de 9 de 2011. [Citado el: 30 de 5 de 2016.] <http://jviezca.blogspot.com/2011/09/tarjetas-crc.html>.
41. esteban sanabria, juan. Calameo. [En línea] [Citado el: 1 de 6 de 2016.] <http://es.calameo.com/books/002879052b8f13f0fcbe9>.
42. Ingeniería De Sistemas (TGS). [En línea] [Citado el: 2 de 6 de 2016.] <http://ingenieroalvarop.blogspot.com/p/caja-negra-y-caja-blanca.html>.
43. Angelfire. [En línea] [Citado el: 2 de 6 de 2016.] <http://www.angelfire.com/empire2/ivansanes/bywbox.htm>.

Referencias

44. Globe. [En línea] [Citado el: 2 de 6 de 2016.] <http://www.globetesting.com/pruebas-de-rendimiento/>.