

**Universidad de Sancti Spíritus
"José Martí Pérez"
Facultad Ingeniería**



**Trabajo de Diploma para optar por el Título de Ingeniero
Informático.**

**Título: Software para contribuir al desarrollo del proceso de
enseñanza-aprendizaje del Álgebra Lineal en la Universidad de
Sancti Spíritus "José Martí Pérez".**

Autor: Yadier Fuentes Garcia.

Tutores: MSc. Tania Rosa Toledo Martín.

Ing. Iosmel Sánchez Martínez.

Sancti Spíritus

Junio 2014

Pensamiento

“Hay que ampliar el campo de los sueños... Porque no se trata solo de la informática para comunicarse, sino para saber, aprender, enseñar, ayudar, compartir.”

Fidel Castro.

Agradecimientos

A mis padres por siempre apoyarme en todas mis decisiones. Por demostrarme que si se puede, por darme tanto amor y cariño, por depositar en mi toda su confianza y estar a mi lado cuando más los necesito.

A mi familia por el apoyo que siempre he recibido de ellos.

A mis tutores por sus buenos consejos durante la realización de este trabajo.

A mi amigo Yoel que lo considero como un hermano más ya que siempre me ha apoyado en todo.

A todos los compañeros de aula por el apoyo durante estos 5 años.

Agradezco a todos los que de una forma u otra contribuyeron y ayudaron en la realización de este trabajo. A todos muchísimas gracias.

Dedicatoria

A mis padres Emilio F. y Gladys G. por ser las personas más importantes de mi vida.

Resumen

La presente investigación surge para contribuir al proceso enseñanza-aprendizaje de la asignatura Álgebra Lineal de la carrera Ingeniería Informática. En la actualidad existen varios software que sirven de apoyo a esta asignatura los cuales agilizan el tiempo en la solución de ejercicios pero las respuestas allí dadas no dan los elementos necesarios para que el estudiante reafirme el marco conceptual de cómo se realizan las diferentes operaciones.

El trabajo de diploma tiene como objetivo desarrollar un software para contribuir al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en la Universidad de Sancti Spíritus “José Martí Pérez”, además se describen todas las etapas pertenecientes a la metodología *Xtreme Programming* (XP) utilizada en el desarrollo del sistema.

El sistema se implementó haciendo uso del *lenguaje de programación java* y Netbeans v7.2.1 como herramienta para su implementación. Se utilizó para la persistencia de los datos SQLite ya que nos permite tener una Estructura Eficiente y Flexible sin depender de un Servidor de SQL (local o remoto) en el momento de trabajar con los datos.

Abstract

This research began to contribute to the teaching and learning of the subject Linear Algebra Computer Engineering career process. At present there are several software that support this subject which speed the time in solving exercises but the answers given there do not give the necessary elements for the student to reaffirm the conceptual framework of how different operations are performed.

The diploma work is to develop a software to contribute to the development of the teaching and learning of linear algebra at the University of Sancti Spiritus "José Martí Pérez" plus all stages belonging to the methodology Xtreme Programming (XP) are described used in developing the system.

The system was implemented using the Java programming language and as a tool for Netbeans v7.2.1 implementation. Was used for the persistence of SQLite as it gives us an efficient and flexible structure without relying on a SQL server (local or remote) when working with the data.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	4
INTRODUCCIÓN.....	4
1.1 CONCEPTOS GENERALES.....	4
1.1.1 <i>Álgebra Lineal.</i>	4
1.1.2 <i>Proceso de Enseñanza-Aprendizaje.</i>	5
1.1.3 <i>Interrelación o uso de las TIC en el proceso de Enseñanza-Aprendizaje del Álgebra Lineal.</i>	8
1.2 MARCO TEÓRICO REFERENCIAL DE LA INVESTIGACIÓN.....	11
1.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	13
1.3.1 <i>Extreme Programming (XP).</i>	14
1.4 GESTOR DE BASES DE DATOS.....	15
1.4.1 <i>MySQL</i>	15
1.4.2 <i>PostgreSQL</i>	17
1.4.3 <i>SQLite</i>	19
1.4.3 <i>Selección del gestor de base datos.</i>	19
1.9 HERRAMIENTAS UTILIZADAS EN LA PROPUESTA DE SOLUCIÓN.	19
1.9.1 <i>NetBeans IDE 7.2.</i>	19
1.10 HERRAMIENTAS CASE.	20
1.10.1 <i>Visual Paradigm</i>	20
1.10.2 <i>Rational Rose Enterprise</i>	20
1.10.3 <i>Selección de la herramienta case</i>	20
CONCLUSIONES PARCIALES.....	21
CAPÍTULO II: PLANIFICACIÓN Y DISEÑO DE LA APLICACIÓN.	22
INTRODUCCIÓN.....	22
2.1 DESCRIPCIÓN DE LOS PROCESOS VINCULADOS AL CAMPO DE ACCIÓN.....	22
2.2 PROPUESTA DE SOLUCIÓN.	22
2.3 FASE PLANIFICACIÓN.	23
2.3.1 <i>Historias de usuario.</i>	23
2.3.2 <i>Plan de entregas.</i>	25
2.3.3 <i>Plan de iteraciones</i>	26
2.3.4 <i>Plan de duración de las iteraciones.</i>	27
2.4 FASE DE DISEÑO.	28
2.4.1 <i>Diagrama de clases persistentes.</i>	28
2.4.2 <i>Tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración).</i>	29
2.5 CONCLUSIONES PARCIALES.	30
CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA	32
INTRODUCCIÓN.....	32
3.1 FASE DE IMPLEMENTACIÓN.	32
3.1.1 <i>Tareas de programación por iteraciones.</i>	32

3.2	FASE DE PRUEBA.....	36
3.2.1	<i>Pruebas de aceptación</i>	36
	CONCLUSIONES PARCIALES.....	42
	CONCLUSIONES GENERALES.....	43
	RECOMENDACIONES.....	44
	BIBLIOGRAFÍA.....	45
	ANEXOS	48

FIGURA # 1: PANTALLA DE PRUEBA AL SISTEMA. 38

FIGURA # 2: PANTALLA DE PRUEBA AL SISTEMA. 39

FIGURA # 3: PANTALLA DE PRUEBA AL SISTEMA. 41

FIGURA # 4: PANTALLA DE PRUEBA AL SISTEMA. 42

TABLA 1: REPRESENTACIÓN DE LA HISTORIA DE USUARIO NRO. 1..... ¡ERROR! MARCADOR NO DEFINIDO.

TABLA 2: REPRESENTACIÓN DE LA HISTORIA DE USUARIO NRO. 2..... ¡ERROR! MARCADOR NO DEFINIDO.

TABLA 3: REPRESENTACIÓN DE LA HISTORIA DE USUARIO NRO. 3..... ¡ERROR! MARCADOR NO DEFINIDO.

TABLA 4: HISTORIA DE USUARIO POR BUNDLE. ¡ERROR! MARCADOR NO DEFINIDO.

TABLA 5: PLAN DE DURACIÓN DE LAS ITERACIONES. 27

TABLA 6: DIAGRAMA DE CLASES PERSISTENTES 28

TABLA 7: TARJETA CRC CLASE EJEMPLOS. 29

TABLA 8: TARJETA CRC CLASE EJERCICIOS..... 29

TABLA 9: TARJETA CRC CLASE INSERTAR MATRIZ. 30

TABLA 10: TARJETA CRC CLASE MATRICES. 30

TABLA 11: TIEMPO REAL DE IMPLEMENTACIÓN POR BUNDLE (ITERACIÓN 1)..... 33

TABLA 12: TAREA DE PROGRAMACIÓN POR HISTORIAS DE USUARIO. (ITERACIÓN 1)..... 33

TABLA 13: TIEMPO REAL DE IMPLEMENTACIÓN POR BUNDLE (ITERACIÓN 2)..... 34

TABLA 14: TAREA DE PROGRAMACIÓN POR HISTORIAS DE USUARIO. (ITERACIÓN 2)..... 34

TABLA 15: TIEMPO REAL DE IMPLEMENTACIÓN POR BUNDLE (ITERACIÓN 3)..... 35

TABLA 16: TAREA DE PROGRAMACIÓN POR HISTORIAS DE USUARIO. (ITERACIÓN 3)..... 35

TABLA 17: PRUEBA DE ACEPTACIÓN DE LA HU "ADICIONAR EJERCICIO" 37

TABLA 18: PRUEBA DE ACEPTACIÓN DE LA HU "ADICIONAR EJEMPLO" ¡ERROR! MARCADOR NO DEFINIDO.

INTRODUCCIÓN

Las actuales generaciones transitan una época matizada por la información donde se habla comúnmente en escenarios educativos diversos del desarrollo de las potencialidades del ser humano, de la autonomía del alumno y de la necesidad de aprender a aprender (López, 2007). El fácil acceso a la información y su distribución por medios electrónicos multiplica el impacto formativo de las instituciones de educación. La mayor interacción entre las comunidades académicas permite un proceso continuo de mejoramiento de la calidad educativa; la apertura a la interacción mundial potencia los procesos de transformación de las instituciones educativas (Almeida, 2007). Estas nuevas realidades a las que se enfrenta la universidad contemporánea, demandan un proceso de enseñanza-aprendizaje más dinámico, donde el profesor estimule, oriente y ayude al estudiante a gestionar el conocimiento, en dependencia de los objetivos curriculares trazados, las características del modelo pedagógico en el cual están inmersos y el desarrollo alcanzado por la sociedad. (Román, 2013)

El trabajo independiente, por parte de los estudiantes, resulta una de las vías para enfrentar estos retos.

Por otra parte, las tecnologías de la información ofrecen oportunidades para replantear a fondo el proceso de transmisión y de construcción del conocimiento y permiten alcanzar, entre otros, los siguientes beneficios: integración de medios (textos, audio, animación y video), interactividad, acceso a grandes cantidades de información, planes y ritmos de trabajo individualizados y respuesta inmediata al progreso del aprendiz. (Núñez, 2007).

En la UNISS para el desarrollo del proceso de enseñanza-aprendizaje de la asignatura de Álgebra Lineal en los estudiantes de Ingeniería Informática, existe experiencia en el uso de software como lo son el Derive y el Mathematica en diferentes versiones. Su uso ha dado ventaja como es el proceso de agilizar el tiempo en la solución de ejercicios, pero que cuando ofrece la respuesta no da elementos necesarios para que el estudiante reafirme el marco conceptual de cómo se realizan las diferentes operaciones, es decir les da la respuesta, introducidos los datos o les permite verificar los cálculos realizados manualmente y en caso de no coincidencia no le ofrece la manera de verificar en donde se ubica el error de procedimiento o cálculo. Además no le posibilita al docente incorporar ejercicios con carácter individual.

Se define por todo lo anterior descrito como **problema a resolver**:

¿Cómo contribuir al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en la Universidad de Sancti Spíritus “José Martí Pérez”?

Para solucionar el problema planteado se propone como **objetivo general**: Elaborar un software para contribuir al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en la Universidad de Sancti Spíritus “José Martí Pérez”.

Para dar cumplimiento al objetivo general se crean las siguientes **preguntas científicas**:

1. ¿Qué fundamentos teóricos y metodológicos sustentan la elaboración de un software que contribuya al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal?
2. ¿Cómo diseñar un software que contribuya al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en la Universidad de Sancti Spíritus “José Martí Pérez”?
3. ¿Cómo implementar un software que contribuya al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en la Universidad de Sancti Spíritus “José Martí Pérez”?

Para responder las preguntas científicas se plantearon las **Tareas de Investigación**:

1. Determinar los fundamentos teóricos y metodológicos que sustentan la elaboración de un software que contribuya al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal.
2. Diseñar un software que contribuya al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en la Universidad de Sancti Spíritus “José Martí Pérez”.
3. Implementar un software que contribuya al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en la Universidad de Sancti Spíritus “José Martí Pérez”.

Una vez culminada las tareas de la investigación se pretende obtener como resultado: la implementación de un software que contribuya al desarrollo del proceso de enseñanza-aprendizaje de la asignatura Álgebra Lineal en los estudiantes de la Universidad José Martí Pérez, contando con una base de datos propia en la cual se almacena toda la información requerida.

El presente trabajo posee tres capítulos:

Capítulo I: Fundamentación Teórica.

En este capítulo se aborda lo referente a los elementos teóricos que soportan la investigación, se presentan las tecnologías y lenguajes de programación que se ajustan al desarrollo del trabajo, fundamentando su selección en base al estudio realizado desde el objeto de estudio al campo de acción.

Capítulo II: Planificación y Diseño.

En este capítulo se realizan a partir de la metodología Extreme Programming (XP) las historias de usuarios generadas por los requerimientos funcionales del sistema, el plan de entrega, iteraciones, reuniones diarias, las tarjetas CRC y el diagrama de clases persistentes pertenecientes a la fase de planificación y diseño brindando así una visión más clara del producto y una mejor comprensión del negocio.

Capítulo III: Implementación y Prueba.

En este capítulo se realizan las tareas de programación por iteraciones en las cuales se analiza el tiempo real de programación por Bundle y las tareas de programación por historias de usuarios. También se realizan las pruebas de aceptación al sistema para verificar el funcionamiento óptimo del mismo.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

Introducción.

En el presente capítulo se abordan los elementos teóricos que sirven de base para la presente investigación, realizando el estudio de los principales conceptos de la gestión de los procesos de trabajo y el marco teórico referencial. Se realiza un análisis de las metodologías, tecnologías y herramientas de desarrollo que se tuvieron en cuenta para seleccionar las que se utilizaron en el desarrollo de la solución propuesta.

1.1 Conceptos Generales.

1.1.1 Álgebra Lineal.

En general, como objeto de la Matemática son consideradas todas las formas y relaciones del mundo real que posean objetivamente tal grado de independencia respecto al contenido, que puede ser totalmente abstraída de este último. Además, no sólo las formas abstraídas de la realidad son objeto de estudio de la matemática sino también aquellas lógicamente posibles, determinadas sobre la base de formas y relaciones ya conocidas. En el Plan de Estudio de la carrera de Ingeniería Informática las disciplinas de matemática tienen un espacio especial, tanto las Superiores como las aplicadas por contribuir a la formación integral de estos profesionales, unas de las asignaturas de la disciplina Matemática es el Álgebra Lineal que es estudiada en el primer semestre del primer año de esta carrera.

Esta contribuye al desarrollo de los fundamentos de la formación de un especialista en Ciencias Técnicas, dado que todo ingeniero considera representaciones técnicas y científicas en términos matemáticos con los cuales reflejan los rasgos cuantitativos de los fenómenos que estudia. De tal modo, el objetivo de esta asignatura es lograr que los ingenieros dominen el aparato matemático (relacionado con los temas que en ella se estudia) que los hagan capaces de modelar y analizar los procesos técnicos, económicos, productivos y científicos, utilizando en ello tanto métodos analíticos como aproximados y haciendo uso eficiente de las técnicas de cómputo. Es decir los estudiantes tienen la posibilidad por los conocimientos que poseen utilizar de forma eficiente y óptima para desarrollarse en el trabajo y estudio independiente de la asignatura el uso de software

que faciliten las operaciones con matrices con etiqueta más personalizada en correspondencia con las necesidades de los estudiantes de esta carrera.

1.1.2 Proceso de enseñanza-aprendizaje.

La comprensión de los procesos de enseñanza- aprendizaje y desarrollo en los alumnos ha estado sujeta a los modelos que la Psicología ha aportado a la Pedagogía, a través de diferentes etapas de su desarrollo como ciencia. Estos modelos han orientado la elaboración de las propuestas curriculares en los diferentes países y épocas históricas, dando lugar a diferentes posturas pedagógicas.

Las formas tradicionales de concebir el proceso de enseñanza-aprendizaje van quedando superadas cada vez más, en gran parte debido al propio peso de las demandas de una realidad histórica que se impone a nivel global. Los nuevos paradigmas educativos se centran en el alumno, sujeto activo que construye de manera muy creadora y personal el conocimiento.

Consideraciones generales sobre el aprendizaje.

A continuación se exponen algunos puntos de partida respecto a los procesos de desarrollo y aprendizaje, que constituyen el referente conceptual y metodológico general de nuestro trabajo (Castellanos y Grueiro, 1996).

La formación y la emergencia del hombre como ser social y cultural es un resultado del proceso de **apropiación** (asimilación) de la experiencia histórico-social, que es transmitida de una a otra generación (Vigotsky, 1987). Esta puede ser considerada como la forma exclusivamente humana de **aprendizaje** y constituye siempre un proceso **interactivo**. Se encuentra mediado por la existencia de una cultura que el sujeto va haciendo suya (a partir del dominio de los objetos, de los modos de actuar, de pensar y de sentir, y del dominio de las capacidades que en aquellos encarnan), y por la existencia de "los otros" (que hace de la comunicación, la cooperación y la actividad conjunta, lo característico de esta forma genérica del aprendizaje). Resulta interesante señalar que, como plantea Pozo (1996), el aprendizaje de la cultura conlleva a su vez a una cultura del aprendizaje. Cada sociedad, cada cultura, crea sus formas específicas de aprendizaje, que pasan a ser sistematizadas y legalizadas en los enfoques y prácticas educativas vigentes.

En esta concepción, el medio o entorno social no es una simple condición que favorece u obstaculiza el aprendizaje: es una parte intrínseca del propio proceso y define su esencia. Partiendo de la teoría de la Escuela Histórico-Cultural sobre el desarrollo psíquico como marco conceptual para la aproximación a los procesos de aprendizaje (Vigotsky, 1987;

Riviere, 1992), categorías como “zona de desarrollo próximo”, “andamiaje o sostén” (Coll, 1992; Day et al., 1985; Wertsch y Kammer, 1992) explicitan la naturaleza de la instrucción o enseñanza y el singular papel de los agentes educativos en esta gestión. Por otra parte, ellas enfatizan el rol vital que desempeña la educación en el proceso de crecimiento y desarrollo intelectual y personal. (Vigotsky, 1982). Lo que el niño puede hacer hoy en cooperación, mañana podrá hacerlo solo. Por lo tanto, el único tipo de instrucción adecuado es el que marcha **delante** del desarrollo y **lo conduce**; debe ser dirigida más a las funciones de maduración que a lo ya maduro.(...) La educación debe estar orientada hacia el futuro, no hacia el pasado. La educación –como se hace explícito en el paradigma histórico-cultural- se concibe como motor del desarrollo. Educación desarrolladora es aquella que conduce al desarrollo, que va delante del mismo (a decir de Vigotsky y sus seguidores)- guiando, orientando, estimulando- que tiene en cuenta el desarrollo actual para ampliar continuamente los límites de la zona de desarrollo próximo o potencial, y por lo tanto, los progresivos niveles de desarrollo del sujeto. La educación desarrolladora promueve y potencia, los aprendizajes desarrolladores.

El aprendizaje ha sido conceptualizado desde muy diversos paradigmas y concepciones. Muchas de ellas han ofrecido, a nuestro entender, una visión unilateral, parcializada, y a veces tergiversada, de este proceso. En trabajos anteriores (Castellanos y Grueiro, 1997) han quedado resumidas algunas las limitaciones que más se han manifestado en el quehacer pedagógico. Así, el aprendizaje ha sido comprendido a veces como un proceso que: a) se encuentra restringido al espacio de la institución escolar (aprendizaje formal), a determinadas etapas exclusivas de la vida (a las que preparan para la vida profesional, adulta); b) que maximiza lo cognitivo, lo intelectual, lo informativo, los **saberes**, sobre lo afectivo-emocional, lo ético y lo vivencial, y el **saber hacer**; c) que se realiza individualmente, aunque, paradójicamente, no se tenga en cuenta o se subvalore al individuo; d) como una vía exclusiva de **socialización**, más que de **individualización, de personalización, de construcción y descubrimiento de la subjetividad**; e) como adquisición de conocimientos, hábitos, destrezas y actitudes para **adaptarse** al medio, más que para **aprender a desarrollarse, a aprender y a crecer**.

En sentido general, la **comprensión de un sujeto activo, reflexivo**, ha estado y está en el centro de los diferentes modelos que sirven de base a la práctica escolar en la actualidad, en todos los casos tienen el propósito de sustituir las actitudes pasivas de los alumnos, aún presentes en las aulas como reflejo de la enseñanza tradicional.

En el caso de nuestro país, el proceso de aprendizaje que asumimos, tiene como sustentos teóricos principales:

- **Nuestras mejores tradiciones educativas** sobre cuyas bases se han realizado las precisiones y enriquecimientos pertinentes a nuestra realidad. Entre sus principales exponentes se encuentran: José Agustín Caballero; Félix Varela Morales; José de la Luz y Caballero; José Martí Pérez; Enrique José Varona; José M. Aguayo; Ramiro Guerra y otros.

En el núcleo de los fundamentos de la enseñanza - aprendizaje en nuestros CES se encuentra el **pensamiento pedagógico de José Martí**, pues en las clases y demás actividades docentes ha de tenerse en cuenta que: el profesor no ha de ser un molde donde los alumnos echan la inteligencia y el carácter para salir con sus lobanillos y jorobas, sino un guía honrado, que enseña de buena fe lo que hay que ver, y explica su pro, lo mismo que el de sus enemigos, para que se fortalezca el carácter de hombre al alumno (T8, 298).

- La teoría pedagógica del **ENFOQUE SOCIO- HISTÓRICO-CULTURAL**, basada en las teorías del gran psicólogo soviético Lev Vigotsky, cuyos postulados fundamentales han servido de base a la mayoría de las investigaciones pedagógicas cubanas de la etapa revolucionaria.

Según este enfoque, el sistema de actividades docentes se concibe de tal manera que el alumno pueda reconstruir el conocimiento, primero en el plano ínter-individual, a través de las dinámicas de grupo y luego en lo intra-individual, mediante la solución de tareas personalizadas, según la **Ley de Doble Formación del Desarrollo** (Vigotsky).

- *¿Cuál es la estructura del proceso de enseñanza – aprendizaje universitario?*

El proceso de enseñanza – aprendizaje es un sistema que posee varios elementos estructurales: **objetivos; contenidos; métodos; formas de organización; medios de enseñanza y formas de evaluación y control**. Las modificaciones en uno de ellos van a provocar cambios en los demás. Al mismo tiempo, la posibilidad real de tales modificaciones depende, en mucho, de cómo los demás elementos del sistema reaccionen ante los contenidos novedosos.

1.1.3 Interrelación o uso de las TIC en el proceso de enseñanza-aprendizaje del Álgebra Lineal.

El desarrollo impetuoso del mundo contemporáneo, caracterizado por el predominio de la globalización neoliberal y la unipolaridad, pone grandes retos ante los sistemas educativos, especialmente a las universidades, llamadas a formar los nuevos profesionales capaces de crear y transformar ese otro mundo posible que el ingenio humano a puesto a nuestro alcance.

En estas nuevas condiciones, el proceso de enseñanza – aprendizaje universitario en Cuba se enfrenta a dos grandes dilemas pedagógicos: por un lado es preciso **lograr una educación verdaderamente humanística** que fomente los valores propios de una ética revolucionaria en lugar de la enajenación neoliberal, cuestión que puede lograrse a través de los nuevos **Programas de la Revolución**, los cuales, como expresara el Comandante en Jefe: convertirán a Cuba en el país más culto del mundo, cuyos hijos serán no solo poseedores de profundos conocimientos profesionales, científicos, técnicos y artísticos, según la profesión de cada cual, y el dominio de varios idiomas, sino también de una alta cultura política, histórica, económica y filosófica que les permitirá comprender y enfrentar los grandes desafíos del futuro (Castro Ruz, F, 2001).

El otro desafío a la Didáctica universitaria es el de **explotar eficazmente las posibilidades que abren las TIC** para el perfeccionamiento de la docencia a través del llamado aprendizaje virtual (elearning), o enseñanza no presencial, donde los roles tradicionales de profesores y estudiantes sufren cambios radicales.

La aplicación de la computación en calidad de elevar el proceso de enseñanza-aprendizaje como medio de enseñanza abre singulares perspectivas para el desarrollo del proceso docente educativo, especialmente para la formación autodidacta de los estudiantes, en cualquier tipo de cursos orientados periódicamente por un profesor. A pesar de estas perspectivas, todavía hoy hacemos un uso mínimo e insuficiente de las posibilidades que el uso de la computación como medio pone ante nosotros, y debe ser tarea de todos, tanto predicar con el ejemplo haciendo uso de ella en las clases, como incidir sobre los docentes de otras carreras ajenas a la informática que se encuentren en el radio de acción, para motivar y asesorar el uso de tan novedoso y poderoso recurso.

Muchos docentes reconocen el papel de la computadora sobre otros medios en el proceso de enseñanza - aprendizaje. Comparándola con otros medios técnicos, como el vídeo o la radio, vemos que ésta aventaja a ambos en su capacidad de interactuar con el estudiante. Esta ventaja, unida a la posibilidad de usar imágenes y sonido, la convierte

en un medio de alta capacidad educativa. Todo ello avala su creciente uso en el proceso pedagógico.

El maestro debe considerar a la computadora como un soporte de la enseñanza que aventaja a otros medios por su alto nivel de interacción. Es decir, no verla solo como una nueva herramienta de apoyo en el aula, sino como aquella que puede transformar los métodos tradicionales de enseñanza, si sus posibilidades se utilizan constructivamente sobre la base de una cultura informática.

Una de las limitaciones presentadas para introducir la computadora en la escuela, ha sido la resistencia de los maestros a utilizar la nueva tecnología. En algunos casos porque no quieren verse desplazados en sus funciones por un equipo, y en otros porque no se sienten capacitados ni pueden estar al día, en relación con el desarrollo acelerado de la informática. No obstante, la computadora se ha expandido rápidamente en el sistema educativo, ya sea como objeto de estudio, como medio de enseñanza o como instrumento de trabajo. En cualquiera de las tres formas, es de vital importancia que el maestro analice cuándo, cómo y en qué momento realmente se justifica su uso.

El profesor es la persona más capacitada para conocer los problemas de su aula, de la asignatura que imparte y la solución de los mismos. El sistema de acciones didácticas consecutivas que organiza para llevar adelante su clase permite la incorporación de diversas técnicas que distinguen la misma clase impartida por dos profesores distintos. Sin dudas, la inserción de la computadora en el proceso docente es tarea del profesor, y solo él decide si a pesar de las limitaciones de un programa, este puede ser utilizado por sus alumnos, o si por el contrario pese a las virtudes que brinda el mismo, no satisface los objetivos a alcanzar en la asignatura.

La enseñanza con computadoras considera la computadora como medio para enseñar y lo asocian a las características del software que se emplea con ese fin. Clasifican el software como tutoriales, entrenadores, simuladores, juegos instructivos, examinadores, ambientes de resolución de problemas, herramientas de enseñanza (incluye procesadores de textos, hojas de cálculo, etc.), sistemas expertos, etc.

El Aprendizaje a través de la computadora: Se centra en la utilización de programas computacionales que faciliten el aprendizaje y desarrollo de habilidades. Entre ellos se encuentran los tutoriales, y los entrenadores o ejercitadores y el aprendizaje con la computadora: En este caso la computadora es una herramienta que facilita el aprendizaje sin tener conocimientos en sí. Ejemplo de este tipo son los juegos

educativos, los procesadores de texto, las hojas de cálculo, las bases de datos, los graficadores, las simulaciones, etc.

Para insertar la computadora en una asignatura en particular es necesario partir de los problemas pedagógicos que se presentan en la misma, por ejemplo, la simulación de un proceso no visible en la práctica, o el limitado tiempo con que cuenta una temática para analizar los resultados que el estudiante debe haber calculado previamente, la poca motivación de los estudiantes por la materia, dificultades con los conocimientos propedéuticos en la asignatura, poca bibliografía para el tratamiento de los contenidos, entre otros.

En consecuencia, el análisis de este punto se divide en dos momentos. El primero ocurre al analizar, con un enfoque sistémico, los componentes del proceso de enseñanza en la asignatura y sus relaciones internas. Un segundo momento implica verificar en que medida la computadora puede, ya sea con software de propósito general o con fines específicos docentes, contribuir a resolver estos problemas o algunos de ellos.

Por tanto, deben analizarse la ubicación de la asignatura en plan de estudio, los objetivos de la asignatura, los contenidos, métodos y medios de enseñanza que se utilizan. También debe tenerse en cuenta en que medida incrementa la calidad del proceso pedagógico al integrar conocimientos, analizar variadas alternativas y fenómenos complejos, entre otros. La aplicación de la propuesta de inserción de la computadora implica la existencia de todos los recursos necesarios para poner en práctica el proyecto, lo que incluye la existencia de los programas y aplicaciones previstas en el proyecto.

Luego, en la concepción de la propuesta de inserción de la computadora y su puesta en práctica deben tenerse en cuenta dos momentos importantes:

La selección entre el software existente, de aquel que satisfaga los requerimientos de calidad y los objetivos trazados en la estrategia de computarización.

Diseñar un software, para que sea elaborado a la medida de las exigencias planteadas.

Una vez determinado los momentos en que se va a emplear la computadora, debe seleccionarse y evaluarse la calidad del software existente relacionado con los temas escogidos, ya sea como herramienta de trabajo o medio de enseñanza, siguiendo una metodología para tales fines, y seleccionar aquellos que satisfagan los objetivos que se han propuesto.

En el contexto del proceso pedagógico profesional la calidad está determinada en cómo se desenvuelve la formación del profesional en el mismo, la orientación que del sistema de actividades que son estructuradas para ir conformando las cualidades,

capacidades y destrezas que caracterizan a un determinado profesional.

Este concepto encierra la filosofía contemporánea respecto a este indicador. La calidad se convierte en un concepto para el fabricante y se evalúa desde la concepción del producto, (incluido las tareas de organización del personal, niveles de responsabilidad) hasta su terminación incluyendo ensayos y mantenimiento. Exige la responsabilidad impartida por todos los funcionarios de la dirección, organización y ejecución del proceso y el control en cada una de las etapas de producción. Estas son típicas en producciones por encargo o productos estándares del mercado.

La ventaja más significativa es que el cliente no se implica en el proceso de producción y su desventaja principal es que los errores detectados a posteriori aumentan considerablemente el costo y en el peor de los casos se descarta su uso.

En el caso particular del software para la educación, la evaluación no debe considerar solo sus cualidades ergonómicas y funcionales sino los factores pedagógicos y psicopedagógicos que justifican su participación en un entorno educativo. (Del Toro). Así como los aspectos que se consideran en la calidad del proceso pedagógico.

Considerando todo lo señalado es que se utiliza en el proceso de enseñanza- aprendizaje del Álgebra Lineal para los Ingenieros Informáticos de la UNISS el software que se propone, facilita al profesor trabajar en función de las capacidades que se van formando como consecuencia de los niveles de actividad y comunicación logrados en el proceso de enseñanza-aprendizaje **motivaciones, intereses, capacidades y cualidades del pensamiento**, (la capacidad para la solución de problemas; desarrollo de la creatividad; nivel de independencia cognoscitiva). En su conjunto, constituyen una expresión del nivel de desarrollo que van alcanzando los educandos en el proceso de enseñanza-aprendizaje de las asignaturas y su fomento abre nuevas zonas de desarrollo del pensamiento de los estudiantes. En la universidad son desarrolladas como resultado del carácter problémico, aplicativo y desarrollador que se confiere a las actividades de enseñanza – aprendizaje fomentado con esta nueva alternativa que es el uso de software para atención diferenciada.

1.2 Marco Teórico referencial de la investigación.

Por supuesto, que la efectividad en el uso de la computadora como medio de enseñanza está estrechamente vinculada con el software que se emplee, los que en la mayoría de los

casos se utilizan para apoyar el estudio de temas específicos, con el fin de reforzar el aprendizaje.

Cada uno de estos programas tienen propósitos específicos, dirigidos a contribuir con el desarrollo de alguno (a veces con más de uno) de los aspectos del proceso docente. Unos pretenden enseñar al alumno un contenido nuevo, otros simulan el desarrollo de un proceso físico, los hay que intentan contribuir al desarrollo de alguna habilidad, intelectual o motora; otros sólo pretenden evaluar los conocimientos del estudiante sobre un determinado contenido.

En dependencia de estas características de los software se ha venido estableciendo una agrupación y una clasificación de los mismos tomando como elemento clasificador la función que realizan dentro del proceso docente. Estas propuestas en las universidades no están distantes en especial en la UNISS para el desarrollo del proceso de enseñanza-aprendizaje de la signatura de Álgebra Lineal para los estudiantes de Ingeniería Informática, en el cual existe experiencia en el uso de otros software como lo son el Derive y el Mathematica en diferentes versiones, pero que su uso ha dado ventaja como es el proceso de agilizar el tiempo en la solución de ejercicios, pero que cuando ofrece la respuesta no da elementos necesarios para que el estudiante reafirme el marco conceptual de cómo se realizan las diferentes operaciones, es decir les da la respuesta introducidos los datos o les permite verificar los cálculos realizados manualmente y en caso de no coincidencia no da le ofrece la manera de verificar en donde se ubica el error de procedimiento o calculo además no le posibilita al docente incorporar ejercicios con carácter individual, es decir para el trabajo independiente de los alumnos y la atención diferenciada, a pesar de que la posibilidad de contar con una base de dato para que le profesor incorpore ejercicios con diferentes grados de dificultad es una vía eficaz para elevar el trabajo diferenciado de los estudiantes en correspondencia con el diagnóstico que se tenga de forma individual y colectiva para contribuir a un buen proceso de evaluación del aprendizaje.

Por estas razones y en correspondencia con los resultados obtenidos en la asignatura Álgebra Lineal en cursos anteriores es que se propone la elaboración y uso de este nuevo software donde se consideran todas las posibilidades que no están dadas en los softwares ya citados para lograr un mejor aprendizaje.

Cuando se utiliza el software que se propone en el proceso de enseñanza-aprendizaje de esta asignatura se garantizan los siguientes aspectos:

- Incrementar la motivación:

El estudiante está más motivado, ya que es un participante activo en una situación de aprendizaje. Puede decirse que es lo idóneo para ampliar esa filosofía de "aprender haciendo".

- Tiene una mejor transferencia del aprendizaje:

La transferencia del aprendizaje se refiere a si las habilidades o conocimientos aprendidos en una situación se aplican a otras situaciones. Esto es lo que usualmente sucede con lo aprendido en una simulación, es decir, después, es bien transferido a una situación real. Además, el estudiante trata con diferentes combinaciones de situaciones, mientras que el libro sólo proporciona información y orientaciones de cómo hacer algo.

- Es más eficiente:

Esto significa que ocurre mayor transferencia por unidad de tiempo de aprendizaje con la simulación de los procesos, ejemplo la multiplicación entre matrices compatibles para esta operación, siendo más eficiente desde el punto de vista de que puede agilizar o aumentar la velocidad de un proceso que, normalmente, para ser observado o realizado, consume una gran cantidad de tiempo. En general, una persona aprende más rápido si los detalles son eliminados al inicio de la instrucción.

- Es Libro o Guía Electrónica:

El software brinda información sobre los contenidos de cada determinado tema, pero donde el usuario selecciona libremente el tema sobre el que se va a informar y el orden en que abordará los diferentes tópicos.

1.3 Metodología de desarrollo de software.

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos las tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso limitar la propia habilidad del equipo para llevar a cabo el proyecto. Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías ágiles, las

cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad (Metodologías Ágiles en el Desarrollo de Software, 2010).

1.3.1 Extreme Programming (XP).

Extreme Programming fue creada por Kent Beck en 1996 y define una manera de reunir a clientes y programadores en un equipo firmemente integrado, con condiciones de trabajo que promueven la comunicación y solución de un problema. Debido a la interacción intensa, funciona mejor para equipos pequeños y medianos requiriendo de una gran colaboración y disciplina.

Extreme Programming se encuentra entre las llamadas Metodologías Ágiles, que surgieron como una alternativa a los procesos de desarrollo de software tradicionales. Sus principios fundamentales son: Realimentación entre el cliente y el equipo de desarrollo. Comunicación fluida entre todos los participantes. Simplicidad en las soluciones implementadas.

Entre las principales características de XP que la diferencian de otras metodologías de desarrollo se encuentran: Es una metodología adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Es una de las metodologías de desarrollo de software con más éxito en la actualidad. Se utiliza en proyectos con equipo de desarrollos pequeños y con plazo de entrega corto. La metodología consiste en una programación rápida o extrema. Una particularidad es que tiene como miembro del equipo al usuario final (Castillo, 2009).

Esta metodología de desarrollo del software está dividida en 4 fases:

1ª Fase: Planificación del proyecto. En esta fase se describen las historias de usuarios las cuales son realizadas con la misma finalidad que los casos de usos pero con algunas diferencias. También se hace necesario la creación de un plan de publicaciones o "Release plan" para indicar las historias de usuarios correspondientes que serán publicadas en las versiones del programa. El proyecto se divide en iteraciones de corta duración y al comenzar cada iteración es donde se seleccionan las historias de usuario definidas anteriormente en el plan. El objetivo principal es mostrar una versión nueva a

cada instante para mostrarle al cliente, saber lo que opina y luego seguir programando de forma que se mantenga la comunicación entre el cliente y los desarrolladores. Para lograr una planificación eficiente es necesario desarrollar reuniones diarias para que los desarrolladores expongan sus problemas, soluciones e ideas de forma conjunta (Castillo, 2009).

2ª Fase: Diseño. Sugiere diseños simples, sencillos y libres de complejidad. Usar glosarios de términos y una correcta especificación de los nombres de métodos y clases que ayude a comprender el diseño y facilite sus posteriores ampliaciones y la reusabilidad del código. Propone además usar la técnica de refactorizar lo que significa mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad (Castillo, 2009).

3ª Fase: Implementación. Para la codificación se utilizan patrones de codificación ya creados y estándares. El modelo de trabajo se realiza usando repositorios de códigos y modelos de desarrollo colectivo (Castillo, 2009).

4ª Fase: Pruebas. Hay que asegurarse de que todo lo que se hace funcione correctamente. Para ello, lo mejor es desarrollar la prueba desde el momento que se conocen los casos de uso (o, según XP, las historias del usuario). Por ello, lo mejor es desarrollar las pruebas antes de generar el código para tener una prueba más objetiva del correcto funcionamiento de éste (Castillo, 2009).

Una vez seleccionada la metodología de desarrollo a utilizar se pasa a definir los lenguajes, tecnologías y herramientas que se utilizarán para el desarrollo del sistema.

1.4 Gestor de Bases de Datos.

1.4.1 MySQL.

MySQL, sistema de gestión de base de datos relacional, multihilo y multiusuario, es open source, además de ser muy rápido, fiable y muy fácil de usar. Entre sus principales características se encuentran (Manual MySQL 5.0, 2011):

Interioridades y portabilidad.

- Escrito en C y en C++.

- Funciona en diferentes plataformas.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Un sistema de reserva de memoria muy rápido basado en threads.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible.

Seguridad.

- Un sistema de privilegios y contraseñas que es muy flexible y seguro, que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

Escalabilidad y límites.

- Soporte a grandes bases de datos.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.

Localización.

- El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.
- Todos los datos se guardan en el conjunto de caracteres elegido.
- Estabilidad de MySQL.
- El diseño de MySQL Server es multicapa, con módulos independientes. Algunos de los últimos módulos se listan a continuación con una indicación de lo bien probados que están (Manual MySQL 5.0, 2011):

- Replicación (Estable).
- Hay grandes grupos de servidores usando replicación en producción, con buenos resultados.
- Tablas InnoDB (Estable).

El motor de almacenamiento transaccional InnoDB es estable y es usado en grandes sistemas de producción con alta carga de trabajo.

- Tablas BDB (Estable).

El código Berkeley DB es muy estable, todavía lo están mejorando con la interfaz del motor de almacenamiento transaccional BDB en MySQL Server.

- MyODBC 3.51 (Estable).

MyODBC 3.51 usa ODBC SDK 3.51 y es usado en sistemas de producción ampliamente. Algunas cuestiones surgidas parecen ser cuestión de las aplicaciones que lo usan e independientes del controlador ODBC o la base de datos subyacente.

1.4.2 PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. A continuación se presentan

algunas de las características más importantes y soportadas por PostgreSQL (Postgresql-Es, 2009):

Generales.

- Es una base de datos 100% ACID.
- Integridad referencial.
- Transacciones anidadas.
- Copias de seguridad en caliente.
- Unicode.

Juegos de caracteres internacionales.

- Multi-Version Concurrency Control (MVCC).
- Múltiples métodos de autenticación.
- Acceso encriptado vía SSL.
- Actualización in-situ integrada.
- Completa documentación.

Licencia BSD.

Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, MacOS X, Solaris, Tru64) y Windows 32/64bit.

- Programación / Desarrollo
- Funciones/procedimientos almacenados en numerosos lenguajes de programación.
- Bloques anónimos de código de procedimientos (sentencias DO).
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tiene disponibles, entre otros, tipos

geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.

- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido).
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.

1.4.3 SQLite.

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. La biblioteca implementa la mayor parte del estándar SQL-92, incluyendo transacciones de base de datos atómicas, consistencia de base de datos, aislamiento, y durabilidad (ACID), triggers y la mayor parte de las consultas complejas. La biblioteca puede ser usada desde programas en C/C++, aunque existen driver de conexión disponibles en otros lenguajes de programación interpretado como Java . (The Definitive Guide tu SQLite, 2010).

1.4.4 Selección del gestor de base datos.

Los gestores de base de datos antes mencionados, presentan características que lo hacen muy buenos para el desarrollo de la solución propuesta, pero entre los requerimientos del cliente se propone usar SQLite ya que nos permite tener una Estructura Eficiente y Flexible sin que nos preocupemos por diseñar alguna por nuestra cuenta, y no dependamos de un Servidor de SQL (local o remoto) en el momento de trabajar con los datos. Por lo que con este gestor el despliegue es mínimo el contrario de los otros dos gestores.

1.5 Herramientas utilizadas en la propuesta de solución.

1.5.1 NetBeans IDE 7.2.

NetBeans IDE es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Soporta lenguajes de programación: Java, PHP, C/C++ entre los más importantes. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso (Netbeans.org, 2012).

1.6 Herramientas Case.

1.6.1 Visual Paradigm.

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Turner, 2013).

1.6.2 Rational Rose Enterprise.

Rational Rose Enterprise es el producto más completo de la familia Rational Rose. Todos los productos Rational Rose incluyen soporte Unified Modeling Language (UML).

Soporta patrones de ANSI C++, Rose J y Visual C++, Enterprise JavaBeans 2.0, e ingeniería directa e inversa para algunas de las construcciones más comunes de Java.

Es capaz de analizar la calidad del código y de generar código gracias a las capacidades de sincronización configurable entre el modelo y el código, además de una gestión más detallada y el uso de modelos con la función de componentes de modelos controlables por separado.

Incluye un complemento de modelado web, que proporciona la capacidad de visualización y el modelado, y herramientas para desarrollar aplicaciones web.

Permite el modelado UML para diseñar bases de datos, con la posibilidad de representar la integración de los requisitos de datos y aplicaciones mediante diseños lógicos y físicos (Rational Rose Enterprise, 2010).

1.6.3 Selección de la herramienta case.

Visual Paradigm es la herramienta case que se utilizará para el desarrollo de la solución, ya que el equipo de trabajo dispone de gran experiencia con la herramienta, además la misma soporta el ciclo de vida completo de desarrollo de software.

Conclusiones Parciales.

Después de realizado el estudio del presente capítulo se arriban a las siguientes conclusiones parciales:

1. Se identificaron los procesos de trabajos como son: procesos claves, estratégico y de apoyo.
2. Se realizó el estudio del marco teórico referencial donde no existe un software que solucione el problema existente en los que se usan actualmente.
3. Se definió la arquitectura para la solución del sistema quedando conformada de la siguiente manera:
 - (XP) como metodología de desarrollo.
 - SQLite como gestor de base de datos.
 - NetBeans IDE 7.2 como entorno de desarrollo.

CAPÍTULO II: PLANIFICACIÓN Y DISEÑO DE LA APLICACIÓN.

Introducción.

Después de haber analizado el marco teórico referencial y elegido las herramientas y metodología a utilizar, se está en condiciones de comenzar el desarrollo del siguiente capítulo. En este capítulo se desarrollan las fases de Planificación y Diseño propias de la metodología propuesta para el desarrollo del sistema así como los artefactos generados en dichas fases.

2.1 Descripción de los procesos vinculados al campo de acción.

En la UNISS para el desarrollo del proceso de enseñanza-aprendizaje de la signatura de Álgebra Lineal en los estudiantes de Ingeniería Informática, existe experiencia en el uso de software como lo son el Derive y el Mathematica en diferentes versiones. Su uso ha dado ventaja como es el proceso de agilizar el tiempo en la solución de ejercicios, pero cuando ofrece la respuesta no da elementos necesarios para que el estudiante reafirme el marco conceptual de cómo se realizan las diferentes operaciones, es decir les da la respuesta introducidos los datos o les permite verificar los cálculos realizados manualmente y en caso de no coincidencia no le ofrece la manera de verificar en donde se ubica el error de procedimiento o cálculo. Además no le posibilita al docente incorporar ejercicios con carácter individual.

2.2 Propuesta de Solución.

Por estas razones y en correspondencia con los resultados obtenidos en la asignatura Álgebra Lineal en cursos anteriores es que se propone la elaboración y uso de un nuevo software donde se consideran todas las posibilidades que no están dadas en los softwares ya citados para lograr un mejor aprendizaje. Ayudando en el trabajo independiente de los alumnos y la atención diferenciada, contando con una base de datos donde el profesor incorpore ejercicios con diferentes grados de dificultad y así elevar el trabajo diferenciado de los estudiantes en correspondencia con el diagnóstico que se tenga de forma individual y colectiva contribuyendo a un buen proceso de evaluación del aprendizaje.

2.3 Fase Planificación.

En esta fase se describen las historias de usuarios las cuales son realizadas con la misma finalidad que los casos de usos pero con algunas diferencias. También se hace necesario la creación de un plan de publicaciones o "Release plan" para indicar las historias de usuarios correspondientes que serán publicadas en las versiones del programa. El proyecto se divide en iteraciones de corta duración y al comenzar cada iteración es donde se seleccionan las historias de usuario definidas anteriormente en el plan. El objetivo principal es mostrar una versión nueva a cada instante al cliente, saber lo que opina y luego continuar programando de forma que se mantenga la comunicación entre el cliente y los desarrolladores. Para lograr una planificación eficiente es necesario desarrollar reuniones diarias donde los desarrolladores expongan sus problemas, soluciones e ideas de forma conjunta (Fases-ProgramacionExtrema, 2012).

2.3.1 Historias de usuario.

En XP la gestión de requisitos se basa en artefactos sencillos que posibilita el avance del equipo de desarrollo, el cliente escribe y prioriza las historias de usuario que expresan las necesidades del sistema. Las historias de usuario tienen el mismo propósito que los casos de uso de las metodologías tradicionales. Siendo descritas de forma conjunta por los clientes y el equipo de desarrollo, tal y como ven ellos las necesidades del sistema. Por tanto serán descripciones cortas y escritas en el lenguaje del usuario, sin terminología técnica. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando comience la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. También conducirán el proceso de creación de los test de aceptación. Durante este proceso se identificaron 9 historias de usuario. A continuación se muestran las historias de usuarios más significativas.

Para definir las historias de usuario utilizamos la siguiente planilla, que contiene todos los datos necesarios para desarrollar la funcionalidad descrita. La misma cuenta con el número de la historia de usuario, los usuarios que tienen acceso a la misma, el nombre, la prioridad que tiene esta historia de usuario en el negocio y su riesgo, los puntos

estimados que son los días que demorará la realización de dicha historia de usuario, cuenta con la iteración a la cual ha sido asignada, aparecen los programadores responsables del desarrollo de la historia de usuario, además de una descripción la cual ejemplifica las funcionalidades de cada historia de usuario, las mismas también cuentan con una observación donde aparecen todos los datos que contiene una historia de usuario.

Tabla 1: Representación de la Historia de Usuario Nro. 1.

Historia de Usuario	
Número: 1	Usuario: Profesor.
Nombre Historia de Usuario: Autenticar usuarios en el sistema.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 1ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite la autenticación de los usuarios contra el fichero correspondiente, si los datos son correctos se le asignan los permisos.	
Observaciones:	

Tabla 2: Representación de la Historia de Usuario Nro. 2.

Historia de Usuario	
Número: 2	Usuario: Profesor.
Nombre Historia de Usuario: Gestionar Ejemplos.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 1ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite gestionar los ejemplos. En el mismo se adicionan, elimina y se listan los ejemplos.	
Observaciones:	

Tabla 3: Representación de la Historia de Usuario Nro. 3.

Historia de Usuario	
Número: 3	Usuario: Profesor.
Nombre Historia de Usuario: Gestionar ejercicio.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 1ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite gestionar los ejercicios. En el mismo se adicionan, elimina y se listan los ejercicios.	
Observaciones:	

(Ver anexo 1 – 9)

Una vez definidas las historias de usuarios, con los grados de complejidad de las mismas daremos continuación al plan de entregas.

2.3.2 Plan de entregas.

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto como son: el desarrollador, los especialistas principales. Se crean los Bundle que implementan una funcionalidad específica. Este plan se acopla a funcionalidades referentes a un mismo tema en Bundle, esto permite un mayor entendimiento en la fase de implementación quedando de la siguiente forma:

Tabla 4: Historia de Usuario por Bundle.

Bundle	Historias de usuarios que abarca.
Usuario.	- Autenticar usuarios en el sistema. - Gestionar usuario.
Profesor	- Gestionar Ejemplos. - Gestionar Ejercicios.

Estudiante	<ul style="list-style-type: none"> - Gestionar Matrices. - Realizar operaciones con matrices. - Realizar ejercicios. - Revisar ejemplos. - Gestionar Ayuda.
------------	--

2.3.3 Plan de iteraciones.

Después de haber identificado las historias de usuarios y realizado una previa planificación, se procede a la planificación de la etapa de implementación del sistema. En este plan se especifica la prioridad con que se irán implementando las historias de usuarios organizadas por iteraciones así como posibles fechas de liberación. Este paso es de suma importancia en el ciclo de desarrollo de XP. Las funcionalidades son planificadas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

1ra Iteración.

En esta iteración se implementarán todas las historias de usuarios que tienen prioridad alta, es decir, las de riesgo 4 y 5, de esta forma se van creando las funcionalidades principales del sistema que dan soporte a la implementación de las demás funcionalidades. Estas historias de usuarios son 1, 2, 3, 8 las cuales hacen alusión de modo general a la autenticación en el sistema como parte de la seguridad de la aplicación y la gestión de los procesos. Además se tendrá una versión de prueba, la cual será mostrada al cliente con el objetivo de obtener una retroalimentación para el grupo de trabajo.

2da Iteración.

El objetivo de esta iteración es la implementación de las restantes funcionalidades con prioridad media, las cuales presentan riesgo 3. Con la culminación de la misma se tendrán

implementadas las peticiones del cliente descritas en las historias de usuario 5, 6 en las cuales se hace mención a funcionalidades como la gestión de matrices, además de realizar algunas de las operaciones con matrices que están definidas. La versión de pruebas de esta iteración junto con la pasada será mostrada al cliente con el objetivo de obtener su valoración y los cambios en el caso de que se encuentren.

3ra Iteración.

En esta iteración serán implementadas las funcionalidades de prioridad baja con riesgo 2. Entre ellas tenemos: gestionar matriz la cual nos permite insertar, eliminar y listar las matrices. Realizar operaciones con matrices, como suma, resta, determinante, entre otras. Estas funciones están descritas en las historias de usuario 4, 7 y 9 respectivamente. Como resultado de esta iteración se tendrá la versión 1.0 del producto final, adicionando lo concerniente a la interfaz. A partir de este momento el sistema será puesto a prueba por un período de tiempo de 2 semanas donde interviene todo el equipo, es decir el programador y los especialistas principales, para evaluar el desempeño del mismo, tomando decisiones y corrigiendo los errores que pudiera presentar el sistema en esta etapa.

2.3.4 Plan de duración de las iteraciones.

Siguiendo el desarrollo de la metodología XP se crea el plan de duración de las iteraciones. En este plan se especifica más detalladamente el orden de desarrollo de las historias de usuarios dentro de cada iteración así como la estimación completa de dicha iteración.

Tabla 5: Plan de duración de las Iteraciones.

Iteraciones.	Orden de las Historias de Usuarios a implementar.	Duración total de las iteraciones.
1ra Iteración.	<ol style="list-style-type: none"> 1. Autenticar usuario en el sistema. 2. Gestionar Ejemplo. 3. Gestionar Ejercicio. 4. Gestionar Usuario. 	8 Semanas

2da Iteración.	<ol style="list-style-type: none"> 1. Revisar ejemplos. 2. Realizar ejercicios. 	7 Semanas
3ra Iteración.	<ol style="list-style-type: none"> 1. Gestionar Matriz. 2. Realizar operaciones con matrices. 3. Gestionar Ayuda. 	5 Semanas

Una vez realizado los artefactos pertenecientes a la fase de planificación a partir de estos se le dará comienzo a la fase de diseño la cual es de gran importancia para el desarrollo del producto.

2.4 Fase de Diseño.

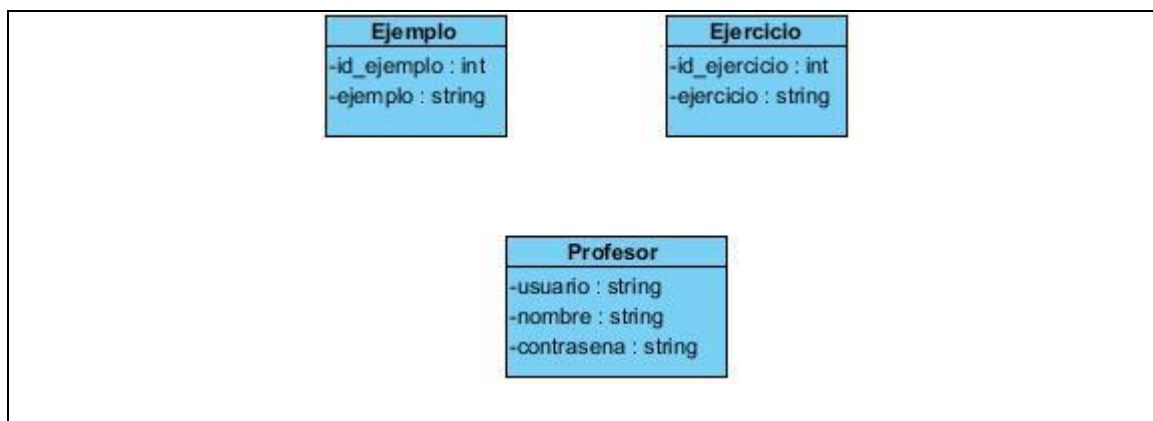
El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos. En esta fase se realiza el diagrama de clases persistentes y las tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración) para una mayor concentración y apariencia del desarrollo al programador (Fases-ProgramacionExtrema, 2012).

2.4.1 Diagrama de Clases persistentes.

En el diagrama de Clases persistentes es donde se muestran las clases persistentes en la base de datos.

Tabla 6: Diagrama de Clases persistentes.

Diagrama de Clases persistentes
--



2.4.2 Tarjetas CRC (Cargo o Clase, Responsabilidad y Colaboración).

Las tarjetas C.R.C representan objetos, la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda aparecen las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

Tabla 7: Tarjeta CRC Clase Ejemplo.

Clase: Ejemplos.	
Responsabilidad.	Colaboración.
Mostrar Ejemplos.	RenderCeldaEjemplo RenderCeldaEjemplo2 RenderCeldaEjemplo3 RenderCeldaNull SQLite Tablas Ejemplo Pos

Tabla 8: Tarjeta CRC Clase Ejercicio.

Clase: Ejercicios.

Responsabilidad.	Colaboración.
Realizar Ejercicios.	Ejercicio RenderCelda RenderCeldaNull SQLite Herramientas Mensajes Tablas

Tabla 9: Tarjeta CRC Clase Insertar Matriz.

Clase: Insertar Matriz.	
Responsabilidad.	Colaboración.
Insertar Matriz.	Mensajes Tablas DefaultTableModel Matrices1 Matriz

Tabla 10: Tarjeta CRC Matrices.

Clase: Matrices.	
Responsabilidad.	Colaboración.
Gestionar Matrices.	Herramientas
Gestionar Usuarios.	Manipular_Fichero
Gestionar Ejercicios.	Mensajes
Gestionar Ejemplos.	

Realizar Cálculos.	Tablas Calculos Matrices1 Matriz EditarProfesor
--------------------	---

Conclusiones Parciales.

Durante el desarrollo de este capítulo se arribaron a las siguientes conclusiones parciales:

1. Se realizó una descripción de los procesos vinculados al campo de acción donde se refleja la complejidad del negocio.
2. En la fase de planificación se generaron los siguientes artefactos:
 - Las historias de usuario.
 - El plan de entrega.
 - Plan de iteraciones.
 - Plan de duración de las iteraciones.
3. En la fase de diseño se obtuvieron los artefactos siguientes:
 - El diagrama de clases persistentes.
 - Las Tarjetas CRC.

CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA

Introducción.

La implementación dentro de la metodología XP debe realizarse de forma iterativa, esta característica trae consigo que después del desarrollo de cada iteración surja un producto funcional que debe ser mostrado al cliente y previamente probado para incrementar la visión de los desarrolladores y clientes de posibles cambios y soluciones. En este capítulo se detallan las tres iteraciones llevadas a cabo durante la construcción del sistema exponiéndose las tareas generadas por las historias de usuarios, así como las pruebas de aceptación efectuadas sobre el sistema.

3.1 Fase de Implementación.

En esta fase XP plantea la implementación de cada una de las historias de usuarios. En un comienzo de la fase se hace un chequeo de cada una de las HU junto con el plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan se crean tareas para ayudar a organizar la implementación exitosa de las HU, asignando al grupo de desarrollo conformado por un desarrollador, los especialistas principales, siendo los responsables de la implementación del sistema. Estas tareas pueden escribirse en un lenguaje técnico, al contrario de las historias de usuario que son escritas en el lenguaje del cliente (Fases-ProgramacionExtrema, 2012).

Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se detallan las tareas de programación por iteraciones.

3.1.1 Tareas de programación por iteraciones.

Cada HU como funcionalidad de la aplicación está compuesta por una o varias tareas de programación, éstas no son más que pasos lógicos a seguir por el programador para realizar la implementación de una HU.

Iteración 1.

En esta iteración se implementaron las historias de usuario de mayor prioridad, que son sin dudas el esqueleto del negocio anteriormente descrito, con el fin de obtener una versión del producto con algunas de las funcionalidades críticas para ser mostrado al cliente y tomar nuevas iniciativas de forma rápida.

Tabla 11: Tiempo real de implementación por bundle (Iteración 1).

Bundle	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
Usuario.	Autenticar usuario del sistema	1	0.2
	Gestionar Usuario.	1	0.4
Profesor.	Gestionar ejemplos.	1	1
	Gestionar ejercicios.	1	1

Tabla 12: Tarea de programación por historias de usuario. (Iteración 1)

Historia de Usuario.	Tarea de programación por historias de usuario.
Autenticar usuario del sistema.	1 - Mostar interfaz "Autenticación". 2 - Autenticación de los usuarios.
Gestionar ejemplos.	1 - Mostrar interfaz "Insertar ejemplos". 2 - Adicionar ejemplos. 3 - Mostrar interfaz "Eliminar ejemplos". 4 - Eliminar ejemplos.
Gestionar ejercicios.	1 - Mostrar interfaz "Insertar ejercicios". 2 - Adicionar ejercicios. 3 - Mostrar interfaz "Eliminar ejercicios". 4 - Eliminar ejercicios.
Gestionar Usuario.	1 - Mostrar interfaz "Gestionar Usuario". 2- Gestionar Usuario.

(Ver anexo 10 – 21)

Iteración 2.

Durante el transcurso de esta iteración se concluye la implementación de los Bundles, debido a que sus funcionalidades fueron divididas en varias iteraciones de manera que no hubiera sobrecarga de trabajo en la 1ra iteración. Las historias de usuarios que se implementan son las de media prioridad.

Tabla 13: Tiempo real de implementación por bundle (Iteración 2).

Bundle	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
Estudiante.	Realizar ejercicios.	0.6	0.7
	Revisar ejemplos.	0.6	1

Tabla 14: Tarea de programación por historias de usuario. (Iteración 2)

Historia de Usuario.	Tarea de programación por historias de usuario.
Realizar ejercicio.	1 - Mostrar interfaz "Ejercicio". 2 - Realizar ejercicio.
Revisar ejemplos.	1 - Mostrar interfaz "Ejemplos". 2 - Revisar ejemplos.

(Ver anexo 22 – 25)

Iteración 3.

En esta iteración se implementan las historias de usuario con baja prioridad. Estas funcionalidades reflejan en la aplicación un ambiente más amigable y cómodo para el cliente.

Tabla 15: Tiempo real de implementación por bundle (Iteración 3).

Bundle	Historias de Usuario	Tiempo de Implementación (Semana)	
		Estimación	Real
Profesor.	Gestionar Ayuda.	0.2	0.3
Estudiante.	Gestionar matrices.	0.4	0.3
	Realizar operaciones con matrices.	0.4	0.7

Tabla 16: Tarea de programación por historias de usuario. (Iteración 3)

Historia de Usuario.	Tarea de programación por historias de usuario.
Gestionar matriz.	1 - Mostrar interfaz "Insertar matriz". 2 - Adicionar matriz. 3 - Eliminar matriz.
Realizar cálculos.	1 - Realizar cálculos.
Gestionar Ayuda.	1 - Mostrar interfaz "Gestionar Ayuda". 2 – Gestionar Ayuda.

(Ver anexo 26 – 31)

Una vez concluido el estudio del primer capítulo y realizada las fases de planificación y diseño correspondientes al siguiente capítulo, así como la implementación del sistema estamos en condiciones de realizar las pruebas de aceptación del mismo.

3.2 Fase de Prueba.

Uno de los pilares fundamentales de XP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados así como el tiempo entre la introducción de éste en el sistema y su detección. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores a la hora de introducir cambios o modificaciones (Allende, 2006; Crispin & House, 2002).

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente (Allende, 2006).

3.2.1 Pruebas de aceptación.

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. Durante las iteraciones las HU seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de éstas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Una HU no se considera completa hasta que no ha pasado por sus pruebas de aceptación (Allende, 2006; Beck, 2000).

Para la realización de las pruebas de aceptación utilizamos la siguiente planilla, la cual contiene un escenario donde se refleja la HU a la cual se le realizarán las pruebas, una descripción de dicha HU, variables que no son más que los campos que presenta la HU, estos campos pueden ser válidos (V) es cuando son introducidos correctamente, inválido (I) es cuando el dato que se entra no es correcto, además cuenta con las respuestas del sistema, que no es más que el resultado que se obtiene de la prueba realizada.

A continuación se muestran las pruebas de aceptación correspondiente a las HU “Adicionar Ejemplo” y “Adicionar Ejercicio”.

Tabla 17: Prueba de aceptación de la HU "Adicionar ejercicio".

Escenario	Descripción	Variable 1	Variable 2	Variable 3	Variable 4	Variable 5	Variable 6	Variable 7	Variable 8	Respuesta del Sistema	
Adicionar ejercicio.	Funcionalidad mediante la cual se adicionan ejercicios al sistema.	V	V	V	V	V	V	V	V	Adiciona un nuevo ejercicio en el sistema.	
		Orden de la Matriz 1.	Matriz 1	Operador.	Orden de la Matriz 2.	Matriz 2	Orden del resultado.	Resultado	Información		
		Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato		
		2 2	2 2	+	2 2	2 2	2 2	4 4	Suma		
			2 2			2 2		4 4			
	Funcionalidad mediante la cual se adicionan ejercicios al sistema.	V	V	V	V	V	V	V	V	V	Muestra un mensaje de error. "Tiene que seleccionar el orden de la matriz." (Figura 2)
		Orden de la Matriz 1.	Matriz 1	Operador.	Orden de la Matriz 2.	Matriz 2	Orden del resultado.	Resultado	Información		
		Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato		
		2	2 2	+	2 2	2 2	2 2	4 4	Suma		
			2 2			2 2		4 4			
Funcionalidad mediante la cual se adicionan ejercicios al sistema.	V	V	V	V	V	V	V	V	V	Muestra un mensaje. "A entrado múltiples puntos o mal uso del carácter (-)." (Figura 3)	
	Orden de la Matriz 1.	Matriz 1	Operador.	Orden de la Matriz 2.	Matriz 2	Orden del resultado.	Resultado	Información			
	Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato			
	2 2	2 2	+	2 2	2 2	2 2	4 4	Suma			
		2 2			2 2		a 4				

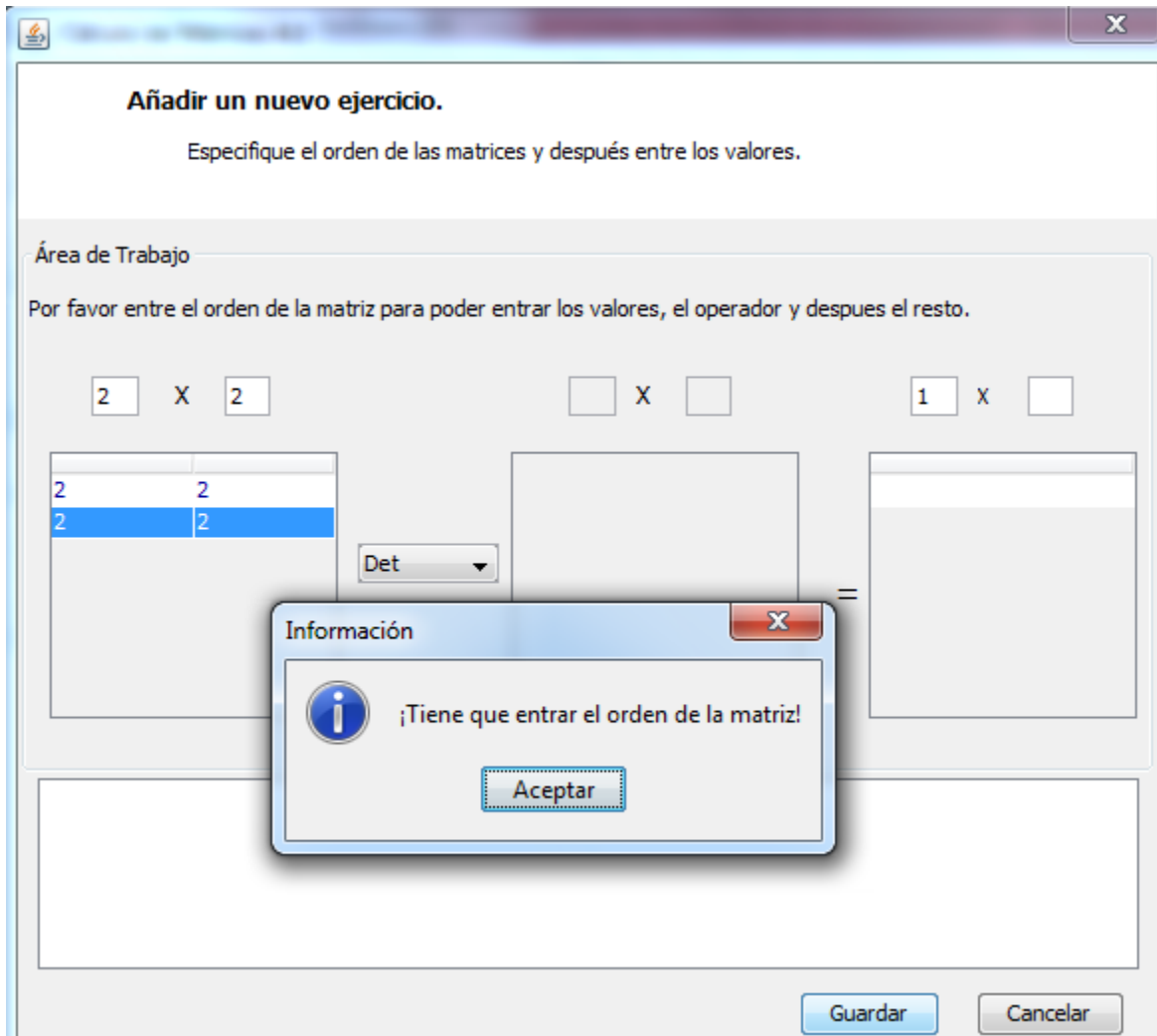


FIGURA # 1: PANTALLA DE PRUEBA AL SISTEMA.

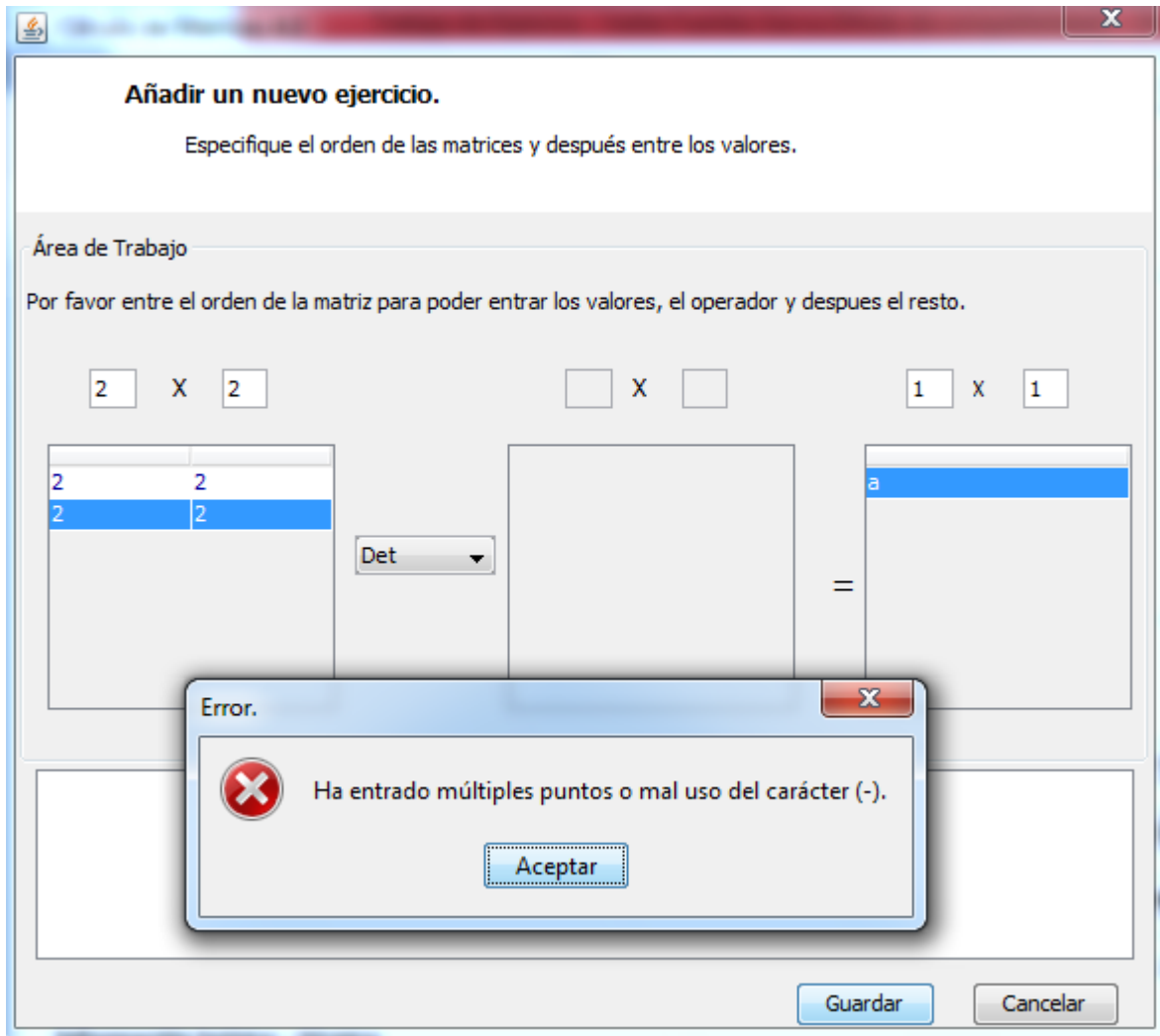


FIGURA # 2: PANTALLA DE PRUEBA AL SISTEMA.

Tabla 18: Prueba de aceptación de la HU "Adicionar ejemplo".

Esce nario	Descripci ón	Variable 1	Varia ble 2	Varia ble 3	Varia ble 4	Varia ble 5	Varia ble 6	Varia ble 7	Varia ble 8	Respuesta del Sistema
Adicio nar ejemp lo.	Funcionali dad mediante la cual se adicionan ejemplos	V Orden de la Matriz 1.	V Matriz 1	V Opera dor.	V Orden de la Matriz 2.	V Matriz 2	V Orden del result ado.	V Result ado	V Inform ación	Adiciona un nuevo ejemplo en el sistema.

al sistema.	Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato	
	2 2	2 2 2 2	+	2 2	2 2 2 2	2 2	4 4 4 4	Suma		
	V	V	V	V	V	V	V	V	V	Muestra un mensaje de error. "Tiene que seleccionar el orden de la matriz."
	Orden de la Matriz 1.	Matriz 1	Operador.	Orden de la Matriz 2.	Matriz 2	Orden del resultado.	Resultado	Información		
	Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato	
2	2 2 2 2	+	2 2	2 2 2 2	2 2	4 4 4 4	Suma		(Figura 4)	
V	V	V	V	V	V	V	V	V	Muestra un mensaje. "A entrado múltiples puntos o mal uso del carácter (-)."	
Orden de la Matriz 1.	Matriz 1	Operador.	Orden de la Matriz 2.	Matriz 2	Orden del resultado.	Resultado	Información			
Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato	Dato		
2 2	2 2 2 2	+	2 2	2 2 2 2	2 2	4 4 a 4	Suma		(Figura 5)	

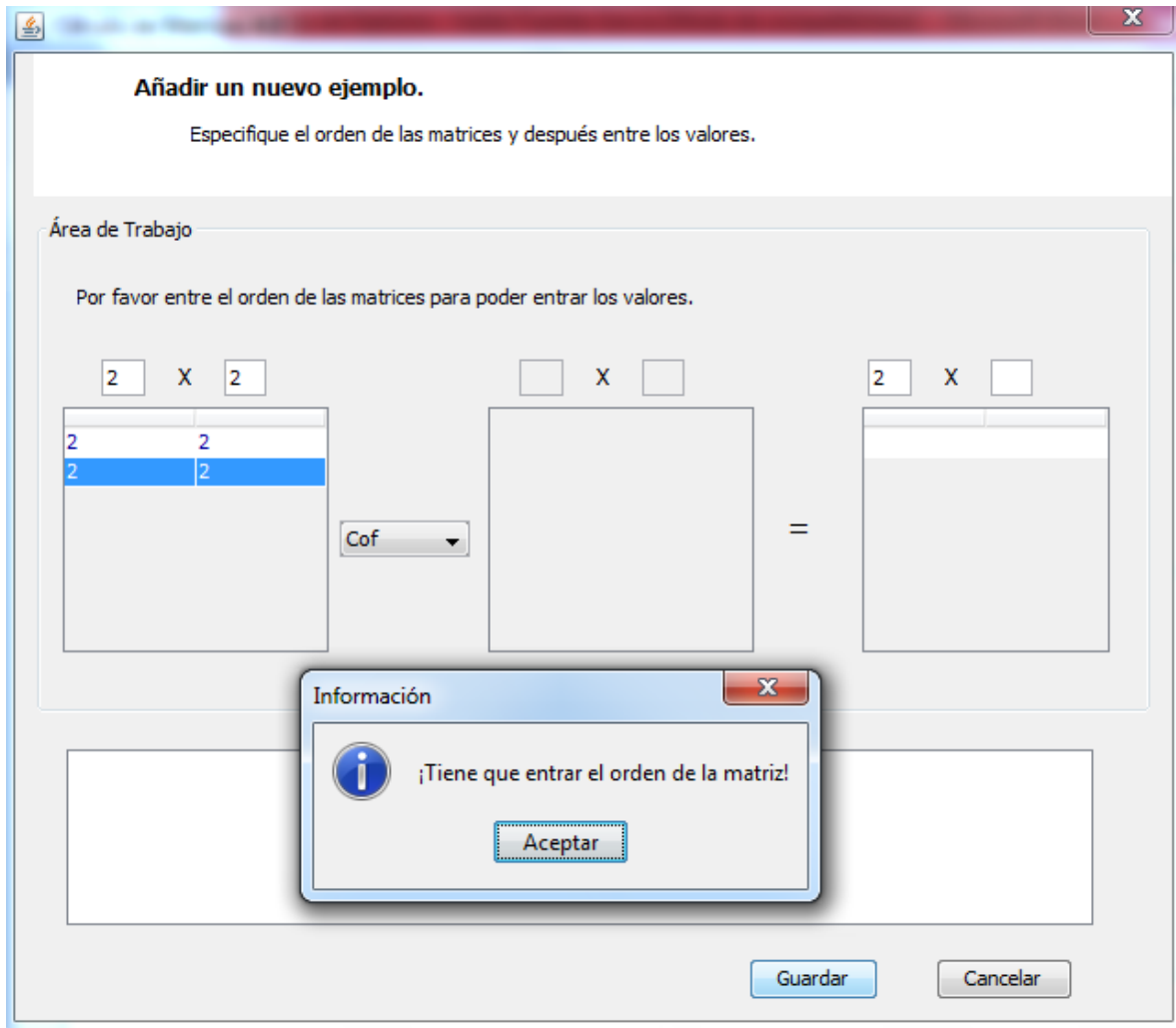


FIGURA # 3: PANTALLA DE PRUEBA AL SISTEMA.

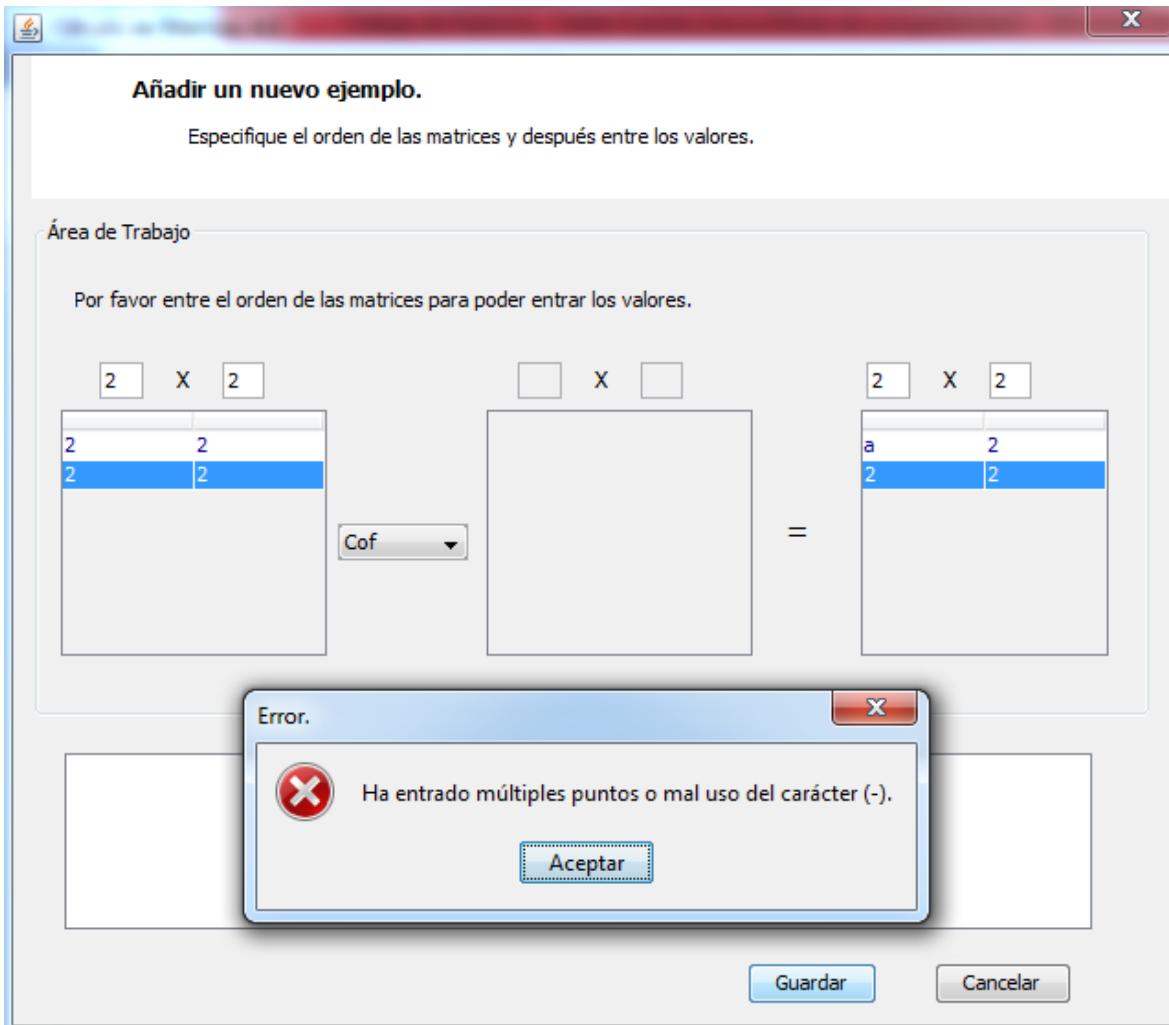


FIGURA # 4: PANTALLA DE PRUEBA AL SISTEMA.

Conclusiones Parciales.

Durante la elaboración de este capítulo se arribaron a las siguientes conclusiones parciales:

1. En la fase de implementación se realizaron las tareas de programación por iteraciones, donde se encuentra las tareas de programación por usuarios y el tiempo real de implementación por Bundle.
2. En la fase de prueba se conformaron los diseños de caso de prueba, realizando las pruebas de aceptación y quedando de esta manera la validación del sistema.

CONCLUSIONES GENERALES

- Se realizó una revisión bibliográfica sobre las distintas herramientas y tecnologías necesarias para el desarrollo de la aplicación. Quedando definida XP como metodología de desarrollo. SQLite como tecnología y como herramienta NetBeans.
- Se planificó un software que contribuye al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en los estudiantes de la Universidad de Sancti Spíritus “José Martí Pérez”.
- Se diseñó un software que contribuye al desarrollo del proceso de enseñanza-aprendizaje del Álgebra Lineal en los estudiantes de la Universidad de Sancti Spíritus “José Martí Pérez”.
- Se implementó un software que contribuye al desarrollo del proceso de enseñanza-aprendizaje de la asignatura Álgebra Lineal en los estudiantes de la Universidad de Sancti Spíritus “José Martí Pérez”.

RECOMENDACIONES

1. Implementar las operaciones con matrices que faltan en los ejercicios y los ejemplos, crear una interfaz más agradable para estos.
2. Modificar la interfaz de la ayuda para que sea más agradable al usuario.

BIBLIOGRAFÍA

Allende, Roberto. (2006) Desarrollo de Portales y Extranet con Plone.

Álvarez, Díaz. (2010). Metodologías Tradicionales vs Metodologías Ágiles.

Beck, K. (2000) .Extreme Programming Explained. s.l.: Addison Wesley.

Castillo, Oswaldo. Extreme Programming. (2009), [Citado: 10 de Febrero del 2013] disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/4302/1/CD-3498.pdf>

Cohn, Mike. (2004). Advantages of User Stories for Requirements. Mountain Goat Software. Octubre del. [Citado: 4 de Febrero del 2013] disponible en: http://www.mountaingoatsoftware.com/article_view/27-advantages-of-user-stories-for-requirements

Crispin, L. y House, T. (2002). Testing Extreme Programming. s.l.: Addison Wesley. [Citado: 27 de Febrero del 2013]

Doris Castellanos Simons .(2000).La comprensión de los procesos del aprendizaje: Apuntes para un marco conceptual, [Citado: 12 de Febrero del 2013]

Extreme Programming. (2006): A gentle introduction. Extreme Programming. [Citado: 4 de Febrero del 2013.] disponible en: <http://www.extremeprogramming.org/>

Fases-ProgramacionExtrema. (2012), [Citado: 22 de Febrero del 2013] disponible en: <http://programacionextrema.tripod.com/fases.htm>

Fowler, Martin. Planning and Running an XP Iteration. Martin Fowler's Blog. 12 de Enero del 2001. [Citado: 4 de Febrero del 2013.] disponible en: <http://martinfowler.com/articles/planningXpIteration.html>

Grant Allen, Mike Owens. (2010), The Definitive Guide tu SQLite. [Citado: 5 de Mayo del 2013]

IIS. (2011), [Citado: 20 de Febrero del 2013] disponible en: <http://www.iis.net/overview>

Jeffries, Ron, Anderson, Ann y Hendrickson, Chet. (2000) .Extreme Programming Installed. : Addison Wesley.

Joskowicz. (2008). Reglas y Practicas en eXtreme Programming.

Lipschutz Seymour.(1979) Álgebra Lineal. Teoría y 600 problemas resueltos. Editorial Pueblo y Educación. [Citado: 21 de Febrero del 2013]

Lowell J. Paige, J. Dean Swift.(1967) Elementos de Álgebra Lineal. Editorial Reverté, [Citado: 21 de Febrero del 2013]

Manual MySQL 5.0. Estabilidad de MySQL (2011), [Citado: 15 de Febrero del 2013] disponible en: <http://dev.mysql.com/doc/refman/5.0/es/stability.html>

Manual MySQL 5.0. (2011). Panorámica del sistema de gestión de base de datos MySQL [Citado: 15 de Febrero del 2013] disponible en: <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>

Manual MySQL 5.0. (2011). Las principales características de MySQL, [Citado: 15 de Febrero del 2013] disponible en: <http://dev.mysql.com/doc/refman/5.0/es/features.html>

Metodologías Ágiles en el Desarrollo de Software. (2010), [Citado: 10 de Febrero del 2013] disponible en: <http://issi.dsic.upv.es/publications/archives/f1069167248521/actas.pdf>

Netbeans.org. (2012), [Citado: 21 de Febrero del 2013] disponible en: http://netbeans.org/index_es.html

Núñez G., Sheremetov L.(2007) Ambiente Computacional de enseñanza-aprendizaje cooperativo personalizado para la educación superior. [Citado: 27 de Febrero del 2014]

PostgreSQL. (2010). [Citado: 13 de Febrero del 2013] disponible en: <http://www.postgresql.org>

PostgreSQL. (2010). Características fundamentales. [Citado: 13 de Febrero del 2013] disponible en: http://www.netpecos.org/docs/mysql_postgres/x15.html

PostgreSQL. (2010). Comparativas. [Citado: 13 de Febrero del 2013] disponible en: http://www.netpecos.org/docs/mysql_postgres/x108.html

Postgresql-Es. (2009), [Citado: 20 de Febrero del 2013] disponible en: http://www.postgresql.org.es/sobre_postgresql

Rational Rose Enterprise. (2010), [Citado: 22 de Febrero del 2013] disponible en: <http://www-142.ibm.com/software/products/es/es/enterprise/>

Ríos L. (2009) Ambiente de enseñanza-aprendizaje inteligente para la Programación Lógica. Tesis presentada en opción al grado científico de Doctor en Ciencias Técnicas. Universidad Central "Marta Abreu" de Las Villas. Cuba.

Rodas Hinostraza, Raúl. (2007). LinuxCentro.net, [Citado: 12 de Febrero del 2013] disponible en: <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>

Román E. (2013) La dirección del trabajo independiente en la Educación Superior Cubana. Conferencia. Universidad de Sancti Spíritus "José Martí Pérez".

Sánchez Jordi. (2006). ¿Qué es un framework?, [Citado: 12 de Febrero del 2013] disponible en: <http://jordisan.net/blog/2006/que-es-un-framework>

Shore, James y Warden, Shane. (2007). The Art of Agile Development. : O'Reilly,

Stephens, Matt y Rosenberg, Doug. (2003). Extreme Programming Refactored: The Case Against XP. : Apress.

Stewart, James.(2006) Cálculo. Trascendentes tempranas. Parte 4. Editorial Félix Varela. La Habana, [Citado: 20 de Febrero del 2014]

Swokowski, Earl.(2009) Cálculo con Geometría Analítica Tomo III. Editorial Félix Varela. La Habana.

Turner, Kevin. (2013). Visual Paradigm for UML Enterprise Edition, [Citado: 21 de Febrero del 2013] disponible en: <http://es.kioskea.net/download/descargar-28127-visual-paradigm-for-uml-enterprise-edition>

Varela Marcelo, María Virginia.(2007) Álgebra Lineal. Editorial Pueblo y Educación. [Citado: 5 de Enero del 2013]

Anexos

Historias de Usuario.

Anexo 1: Representación de la Historia de Usuario Nro. 1.

Historia de Usuario	
Número: 1	Usuario: Profesor.
Nombre Historia de Usuario: Autenticar usuarios en el sistema.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 1ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite la autenticación de los usuarios contra el fichero correspondiente, si los datos son correctos se le asignan los permisos.	
Observaciones:	

Anexo 2: Representación de la Historia de Usuario Nro. 2.

Historia de Usuario	
Número: 2	Usuario: Profesor.
Nombre Historia de Usuario: Gestionar Ejemplos.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 1ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite gestionar los ejemplos. En el mismo se adicionan, elimina y se listan los ejemplos.	
Observaciones:	

Anexo 3: Representación de la Historia de Usuario Nro. 3.

Historia de Usuario	
Número: 3	Usuario: Profesor.
Nombre Historia de Usuario: Gestionar ejercicio.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 1ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite gestionar los ejercicios. En el mismo se adicionan, elimina y se listan los ejercicios.	
Observaciones:	

Anexo 4: Representación de la Historia de Usuario Nro. 4.

Historia de Usuario	
Número: 4	Usuario: Profesor y Estudiante.
Nombre Historia de Usuario: Gestionar Matriz.	
Prioridad en Negocio: Baja.	Riesgo en Desarrollo: 2
Puntos Estimados: 0.4	Iteración Asignada: 3ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite gestionar las matrices. En el mismo se adicionan, elimina y se listan las matrices.	
Observaciones:	

Anexo 5: Representación de la Historia de Usuario Nro. 5.

Historia de Usuario	
Número: 5	Usuario: Profesor y Estudiante.
Nombre Historia de Usuario: Revisar ejemplos.	
Prioridad en Negocio: Media.	Riesgo en Desarrollo: 3
Puntos Estimados: 0.6	Iteración Asignada: 2da.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite revisar los ejemplos. En el mismo el estudiante puede ver la solución del ejemplo paso a paso.	
Observaciones:	

Anexo 6: Representación de la Historia de Usuario Nro. 6.

Historia de Usuario	
Número: 6	Usuario: Profesor y Estudiante.
Nombre Historia de Usuario: Realizar ejercicio.	
Prioridad en Negocio: Media.	Riesgo en Desarrollo: 3
Puntos Estimados: 0.6	Iteración Asignada: 2da.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite realizar los ejercicios. En el mismo el estudiante puede realizar los ejercicios y comprobar su resultado.	
Observaciones:	

Anexo 7: Representación de la Historia de Usuario Nro. 7.

Historia de Usuario	
Número: 7	Usuario: Profesor y Estudiante.
Nombre Historia de Usuario: Realizar cálculos.	
Prioridad en Negocio: Baja.	Riesgo en Desarrollo: 2
Puntos Estimados: 0.4	Iteración Asignada: 3ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite realizar todas las operaciones con matrices definidas en el software.	
Observaciones: Las operaciones con matrices definidas en el software son : Suma, resta, multiplicación, igualdad, potencia, multiplicación de un número natural por una matriz, determinante, adjunta, rango, inversa, cofactores, traspuesta y matriz escalón. Además de una opción adicional donde se muestran las propiedades de la matriz (Orden, Determinante, Singular, Clasificación y si es inversible).	

Anexo 8: Representación de la Historia de Usuario Nro. 8.

Historia de Usuario	
Número: 8	Usuario: Profesor.
Nombre Historia de Usuario: Gestionar Profesor.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 1ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite gestionar los profesores. En el mismo se modifican los datos del profesor.	
Observaciones:	

Anexo 9: Representación de la Historia de Usuario Nro. 9.

Historia de Usuario	
Número: 9	Usuario: Profesor.
Nombre Historia de Usuario: Gestionar Ayuda.	
Prioridad en Negocio: Baja.	Riesgo en Desarrollo: 5
Puntos Estimados: 1	Iteración Asignada: 3ra.
Programadores Responsables: Yadier Fuentes Garcia.	
Descripción: Permite gestionar la ayuda. En el mismo se modofican, se insertan y se eliminan los temas de la ayuda.	
Observaciones:	

Tareas de programación**Iteración 1.**

Anexo 10: Tarea de Programación: “Mostrar interfaz Autenticación”

Tarea.	
Nro. De Tarea: 1	Nro. De HU: 1 (Autenticar usuarios en el sistema).
Nombre de la Tarea: Mostar interfaz “Autenticación”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.3
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario accede al sistema, este selecciona la opción de autenticarse en el sistema e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 11: Tarea de Programación: “Autenticar usuario del sistema”

Tarea.	
Nro. De Tarea: 2	Nro. De HU: 1 (Autenticar usuarios en el sistema).
Nombre de la Tarea: Autenticar usuario del sistema.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Autenticar usuario del sistema” el usuario inserta usuario y contraseña y selecciona el botón aceptar.	

Anexo 12: Tarea de Programación: “Mostrar interfaz Insertar ejemplos”.

Tarea.	
Nro. De Tarea: 3	Nro. De HU: 2 (Gestionar ejemplos).
Nombre de la Tarea: Mostrar interfaz “Insertar ejemplos”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario se autentifica, este selecciona la opción Insertar ejemplos e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 13: Tarea de Programación: “Adicionar ejemplos”.

Tarea.	
Nro. De Tarea: 4	Nro. De HU: 2 (Gestionar ejemplos).
Nombre de la Tarea: Adicionar ejemplos.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.3
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Insertar ejemplos” el usuario inserta los datos necesarios para adicionar un nuevo ejemplo y selecciona el botón aceptar. Si no ocurren errores el sistema limpia los datos del ejemplo.” De lo contrario muestra un mensaje correspondiente.	

Anexo 14: Tarea de Programación: “Mostrar interfaz eliminar ejemplos”.

Tarea.	
Nro. De Tarea: 5	Nro. De HU: 2 (Gestionar ejemplos).
Nombre de la Tarea: Mostrar interfaz “Eliminar ejemplos”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.2
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario se autentifica, este selecciona la opción eliminar ejemplos e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 15: Tarea de Programación: “Eliminar ejemplo”

Tarea.	
Nro. De Tarea: 6	Nro. De HU: 6 (Gestionar ejemplo).
Nombre de la Tarea: Eliminar ejemplos.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.2
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Eliminar ejemplos” el usuario procede a seleccionar el ejemplo que desea eliminar y selecciona el botón aceptar. Si no ocurren errores el sistema elimina el ejemplo de la lista.” De lo contrario muestra un mensaje adecuado.	

Anexo 16: Tarea de Programación: “Mostrar interfaz Insertar ejercicios”.

Tarea.	
Nro. De Tarea: 7	Nro. De HU: 3 (Gestionar ejercicios).
Nombre de la Tarea: Mostrar interfaz “Insertar ejercicios”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario se autentifica, este selecciona la opción Insertar ejercicios e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 17: Tarea de Programación: “Adicionar ejercicios”.

Tarea.	
Nro. De Tarea: 8	Nro. De HU: 3 (Gestionar ejercicios).
Nombre de la Tarea: Adicionar ejercicios.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.3
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Insertar ejercicios” el usuario inserta los datos necesarios para adicionar un nuevo ejercicios y selecciona el botón aceptar. Si no ocurren errores el sistema limpia los datos del ejercicios.” De lo contrario muestra un mensaje correspondiente.	

Anexo 18: Tarea de Programación: “Mostrar interfaz eliminar ejercicios”.

Tarea.	
Nro. De Tarea: 9	Nro. De HU: 3 (Gestionar ejercicios).
Nombre de la Tarea: Mostrar interfaz “Eliminar ejercicios”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.2
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario se autentifica, este selecciona la opción eliminar ejercicios e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 19: Tarea de Programación: “Eliminar ejercicios”

Tarea.	
Nro. De Tarea: 10	Nro. De HU: 3 (Gestionar ejercicios).
Nombre de la Tarea: Eliminar ejercicios.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.2
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Eliminar ejercicios” el usuario procede a seleccionar el ejemplo que desea eliminar y selecciona el botón aceptar. Si no ocurren errores el sistema elimina el	

ejemplo de la lista.” De lo contrario muestra un mensaje adecuado.

Anexo 20: Tarea de Programación: “Mostrar interfaz Gestionar Usuario”

Tarea.	
Nro. De Tarea: 11	Nro. De HU: 8 (Gestionar Usuario).
Nombre de la Tarea: Mostrar Interfaz “Gestionar Usuario”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario se autentifica, este selecciona la opción gestionar usuario e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 21: Tarea de Programación: “Gestionar Usuario”

Tarea.	
Nro. De Tarea: 12	Nro. De HU: 8 (Gestionar Usuario).
Nombre de la Tarea: Gestionar Usuario.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Gestionar Usuario” el usuario inserta los datos necesarios y selecciona el botón aceptar. Si no ocurren errores el sistema limpia los datos del usuario. De lo contrario muestra un mensaje correspondiente.	

Iteración 2.**Anexo 22: Tarea de Programación: “Mostrar interfaz Realizar ejercicios”.**

Tarea.	
Nro. De Tarea: 13	Nro. De HU: 6(Realizar ejercicios).
Nombre de la Tarea: Mostrar interfaz “Ejercicios”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario accede al sistema, este selecciona la opción Ejercicios e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 23: Tarea de Programación: “Realizar ejercicios”.

Tarea.	
Nro. De Tarea: 14	Nro. De HU: 6 (Realizar ejercicios).
Nombre de la Tarea: Realizar ejercicios.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.3
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Ejercicios” el usuario inserta los datos necesarios para la respuesta del ejercicio. Dependiendo de si la respuesta está bien o no se muestra el mensaje correspondiente.	

Anexo 24: Tarea de Programación: “Mostrar interfaz Revisar ejemplos”.

Tarea.	
Nro. De Tarea: 15	Nro. De HU: 5 (Revisar ejemplos).
Nombre de la Tarea: Mostrar interfaz “Ejemplos”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	

Descripción: Una vez que el usuario accede al sistema, este selecciona la opción Ejemplos e inmediatamente se muestra la interfaz de usuario correspondiente.

Anexo 25: Tarea de Programación: “revisar ejemplos”.

Tarea.	
Nro. De Tarea: 16	Nro. De HU: 5 (Revisar ejemplos).
Nombre de la Tarea: Revisar ejemplos.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.3
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Ejemplos” el usuario puede ver paso por paso la solución de ejemplo.	

Iteración 3.

Anexo 26: Tarea de Programación: “Mostrar interfaz Insertar Matriz”.

Tarea.	
Nro. De Tarea: 17	Nro. De HU: 4 (Gestionar matriz).
Nombre de la Tarea: Mostrar interfaz “Insertar matriz”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.7
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario accede al sistema, este selecciona la opción Insertar Matriz e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 27: Tarea de Programación: “Adicionar Matriz”.

Tarea.	
Nro. De Tarea: 18	Nro. De HU: 4 (Gestionar matriz).
Nombre de la Tarea: Adicionar matriz.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.3
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Insertar matriz” el usuario inserta los datos necesarios para adicionar una nueva matriz y selecciona el botón aceptar. Si no ocurren errores el sistema limpia los datos de la matriz.” De lo contrario muestra un mensaje correspondiente.	

Anexo 28: Tarea de Programación: “Eliminar matriz”

Tarea.	
Nro. De Tarea: 19	Nro. De HU: 4 (Gestionar matriz).
Nombre de la Tarea: Eliminar matriz.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.2
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario accede al sistema el usuario procede a seleccionar en la lista de matrices la que desea eliminar y selecciona el botón eliminar. Si no ocurren errores el sistema elimina el ejemplo de la lista.” De lo contrario muestra un mensaje adecuado. Solo se pueden eliminar las matrices insertadas por el usuario.	

Anexo 29: Tarea de Programación: “Realizar cálculo”

Tarea.	
Nro. De Tarea: 20	Nro. De HU: 7 (Realizar cálculo).
Nombre de la Tarea: Realizar cálculo.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.2
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario accede al sistema el usuario procede a seleccionar en la lista	

de matrices las matrices con que desea trabajar y después procede a seleccionar el cálculo. Si no se puede realizar el cálculo muestra un mensaje adecuado. De lo contrario se muestra el resultado.

Anexo 30: Tarea de Programación: “Mostrar interfaz Gestionar ayuda”.

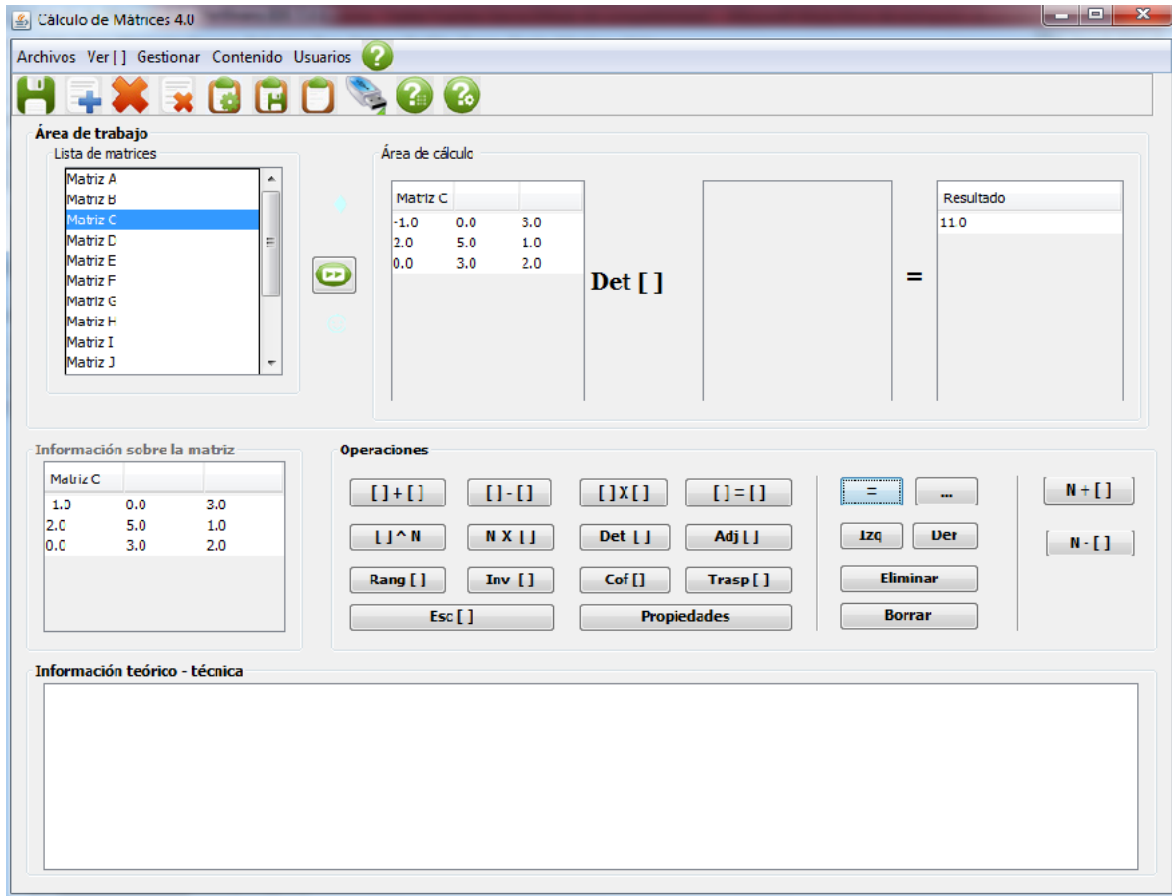
Tarea.	
Nro. De Tarea: 21	Nro. De HU: 9 (Gestionar ayuda).
Nombre de la Tarea: Mostrar interfaz “Gestionar Ayuda”.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.2
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez que el usuario se autentifica, este selecciona la opción Gestionar Ayuda e inmediatamente se muestra la interfaz de usuario correspondiente.	

Anexo 31: Tarea de Programación: “Gestionar Ayuda”.

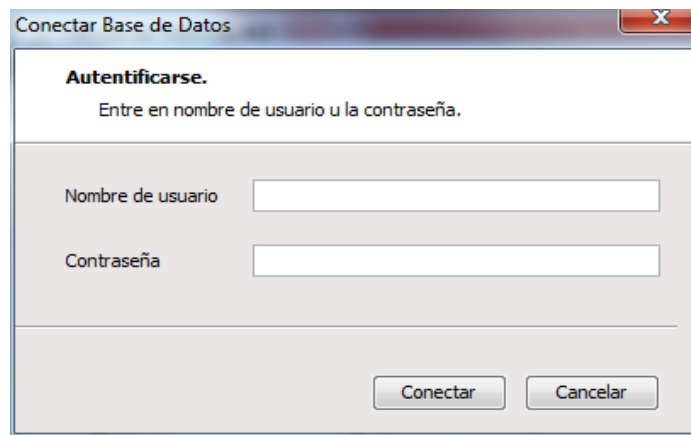
Tarea.	
Nro. De Tarea: 22	Nro. De HU: 9 (Gestionar ayuda).
Nombre de la Tarea: Gestionar Ayuda.	
Tipo de Tarea: Desarrollo.	Puntos de Estimación: 0.3
Programador Responsable: Yadier Fuentes Garcia.	
Descripción: Una vez se muestra la interfaz “Gestionar Ayuda” el usuario adiciona o elimina un tema de la ayuda y selecciona el botón aceptar. El sistema muestra un mensaje correspondiente.	

Pantallas del Sistema.

Anexo 28: Pantalla Principal.



Anexo 29: Pantalla de autenticación.



Anexo 30: Pantalla Insertar matriz.

Añadir una nueva matriz.
Especifique el orden y después entre los valores.

Por favor entre el orden de la matriz para poder entrar los valores.

Orden X

Entre los valores de la matriz en la tabla y después de Guardar.

Anexo 31: Pantalla Insertar Ejercicios.

Añadir un nuevo ejercicio.
Especifique el orden de las matrices y después entre los valores.

Área de Trabajo

Por favor entre el orden de la matriz para poder entrar los valores, el operador y después el resto.

X X X

 =

Operador ▼

Anexo 32: Pantalla Insertar ejemplos.

Añadir un nuevo ejemplo.
Especifique el orden de las matrices y después entre los valores.

Área de Trabajo
Por favor entre el orden de las matrices para poder entrar los valores.

X X X

 =

Operador ▼

Guardar Cancelar

Anexo 33: Pantalla Eliminar ejercicios.

Eliminar Ejercicios.
Tiene que seleccionar un ejercicio en la lista y después eliminarlo.

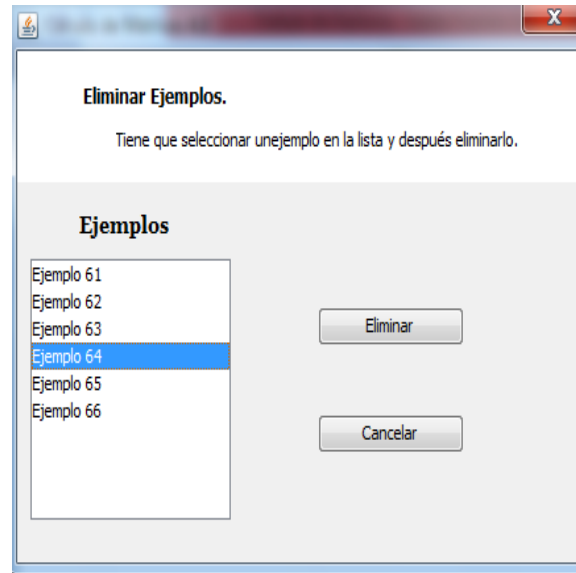
Ejercicios

Ejercicio 1
Ejercicio 2
Ejercicio 3
Ejercicio 4

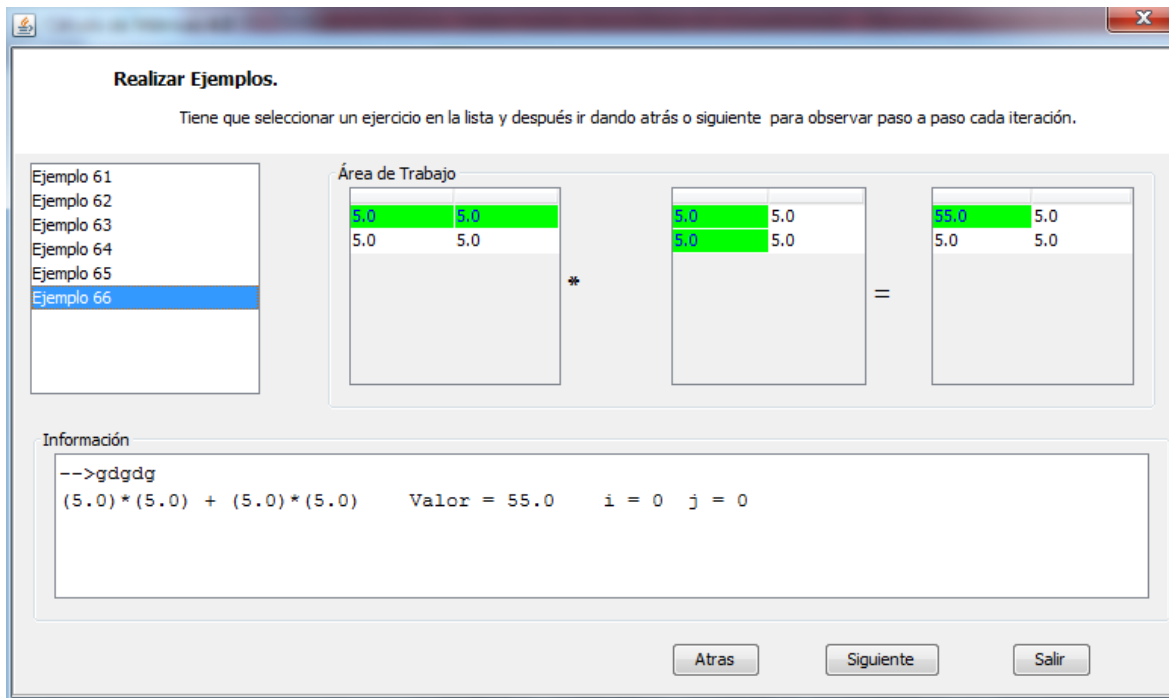
Eliminar

Cancelar

Anexo 34: Pantalla Eliminar Ejemplos.



Anexo 35: Pantalla Ejemplos.



Anexo 36: Pantalla Ejercicios.

Realizar Ejercicios.

Tiene que seleccionar un ejercicio en la lista y después entrar el posible resultado.

Ejercicio 1
Ejercicio 2
Ejercicio 3
Ejercicio 4

Área de Trabajo

1.0	1.0
1.0	1.0

 -

1.0	1.0
1.0	1.0

 =

Anexo 37: Pantalla Editar Profesor.

Profesores

Profesor

Nombre : Yadier Fuenetes

Usuario : Profesor

Password :

New Password :

Chek Password :

Guardar Cancelar

Anexo 38: Pantalla Ayuda.

