

**UNIVERSIDAD DE GUANTÁNAMO**  
**Facultad de Ingeniería y Ciencias Técnicas**

**DISEÑO DEL SISTEMA INFORMÁTICO PARA EL TRABAJO CON MATRICES  
EN LA ASIGNATURA ÁLGEBRA LINEAL DE LA CARRERA INGENIERÍA  
INFORMÁTICA EN LA UNIVERSIDAD DE GUANTÁNAMO**

**Tesis presentada en opción al Título de Ingeniero Informático.**

**Autor:**

Eugeilis Lafita Hernández

**Tutora:**

Ing. Yelenne Marcelo Luis

**Guantánamo, Julio 2020**

**Cuba**

## **DEDICATORIA**

Dedico este trabajo a mí madre y a mí padrastro por siempre confiar en mí, a mis tutores, profesores, compañeros de aula y de trabajo por toda la ayuda que me han brindado a lo largo de mi carrera y en la realización de este proyecto de tesis, así como a Aislen Veranes Pelegrin mi compañera de la vida la cual con su apoyo constante ayudó a la realización de este sueño. A la bujía de mi vida mi niña Stephany la que con tanto amor y cariño eliminó horas de cansancio y estrés que apenas parecían perceptibles con su presencia. A mis amigos, compañeros, profesores y a todos los que de una forma u otra pusieron su grano de arena para que esto pudiera hacerse realidad les dedico estas líneas de agradecimiento.

## **AGRADECIMIENTOS**

Agradezco a mi madre por siempre estar ahí para mí y ser el motor impulsor para que hoy esté optando por el Título de Ingeniero en Informática, por haber confiado en mí y apoyado en cada momento difícil de mis años de estudio, a mis compañeros de aula que nunca me dejaron atrás. Agradezco a mi esposa que siempre me apoyó y nunca dejó que los obstáculos se interpusieran en mi carrera, por haber buscado siempre una solución a los problemas sin afectarme, por haberme regalado tantos momentos de felicidad, apoyo y comprensión, así como por haberme regalado la pequeña más hermosa a que un padre puede aspirar. A alguien muy especial que más que amigo es hermano y que me acompañó durante los seis años de estudio siendo en ocasiones el último recurso, pero el más seguro, gracias Ariolkis.

## RESUMEN

El uso de las Tecnologías de la Información y las Comunicaciones en la educación ha alcanzado mucho auge, su implementación en el proceso de enseñanza-aprendizaje es crucial. Un diagnóstico efectuado a los softwares educativos que se utilizan en la carrera de Ingeniería Informática e investigaciones actuales confirman que existen limitaciones en la aprehensión de los contenidos de sistemas de ecuaciones lineales y matrices en la asignatura de Álgebra Lineal en los estudiantes del 1er año de la carrera de Ingeniería Informática. Para dar solución al problema planteado, se propone como objetivo: diseñar un sistema informático para el aprendizaje del trabajo con matrices de la asignatura Álgebra Lineal en la carrera de Ingeniería Informática de la Universidad de Guantánamo. Obteniendo como resultado un tutorial para el aprendizaje del trabajo con matrices y visualizar paso a paso la solución del ejercicio, sustentado en los referentes teóricos de la comunicación, el desarrollo con facilidades para la enseñanza semipresencial y modular con un alto componente colaborativo. El software se diseñará con tecnologías de: HTML 5, CSS 3, JavaScript, Bootstrap, jQuery y NetBeans.

**Palabras Clave:** Software educativo; Tutorial; Matrices; Enseñanzas matrices; Álgebra lineal.

## **ABSTRACT**

The use of Information Technologies and the Communications in education has caught up with a lot of prosperity, his implementation in the process of teaching learning is crucial. A diagnosis once the educational software was made that they utilize themselves in the race of Informatics Engineering and present-day investigations confirm that limitations in the apprehension of the contents of linear systems of equations and wombs in the subject of study of linear algebra in the students of the 1st year of the Computer Engineering degree. In order to give solution to the presented problem, the authors set for themselves as objective: Developing an information-technology system for the learning of the work with wombs of the subject of study linear algebra in the race of Information Engineering of Guantanamo's University. Getting as a result an appraising instruction manual for the learning of the work with wombs and visualizing the solution of exercise step by step, held in the referent theoreticians of communication, the development with facilities for the semi-eyewitness and modular teaching with a component collaborative height. The software was developed with HTML 5, CSS itself 3, JavaScript, Bootstrap, jQuery and NetBeans.

**Key words:** Educational software; Instruction manual; Matrices; Teaching matrices; Linear algebra.

.

## ÍNDICE GENERAL

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	7
<b>1.1 Introducción al capítulo.....</b>	<b>7</b>
<b>1.2 Análisis y surgimiento del Álgebra Lineal. ....</b>	<b>7</b>
<b>1.2.1 El papel de la historia del Álgebra Lineal en los Procesos de Enseñanza-Aprendizaje.....</b>	<b>8</b>
<b>1.3 Caracterización del proceso de enseñanza-aprendizaje de la asignatura Álgebra Lineal en la carrera de Ingeniería Informática.....</b>	<b>10</b>
<b>1.4 Software Educativo.....</b>	<b>13</b>
<b>1.4.1 Tipología del software educativo. ....</b>	<b>14</b>
<b>1.5 Sistemas informáticos similares. ....</b>	<b>17</b>
<b>1.6 Matrices.....</b>	<b>21</b>
<b>1.6.1 Concepto de matriz.....</b>	<b>21</b>
<b>1.6.2 Tipos de matrices. ....</b>	<b>22</b>
<b>1.6.3 Operaciones con matrices.....</b>	<b>25</b>
<b>1.7 Metodología de desarrollo de software.....</b>	<b>30</b>
<b>1.7.1 Extreme Programming. ....</b>	<b>31</b>
<b>1.8 Lenguajes, herramientas y tecnologías para el desarrollo del sistema informático. ....</b>	<b>33</b>
<b>1.8.1 Aplicaciones de escritorio. ....</b>	<b>33</b>
<b>1.8.2 Aplicaciones web.....</b>	<b>34</b>
<b>1.8.3 Lenguajes de programación web.....</b>	<b>35</b>
<b>1.8.4 Framework del lado del cliente.....</b>	<b>38</b>
<b>1.8.5 Entorno de desarrollo integrado. ....</b>	<b>39</b>
CAPÍTULO 2: DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.....	41
<b>2.1 Introducción del capítulo.....</b>	<b>41</b>
<b>2.2 Personas relacionadas con el sistema. ....</b>	<b>41</b>

<b>2.3 Exploración.....</b>	<b>41</b>
<b>2.4 Restricciones que el sistema debe cumplir.....</b>	<b>43</b>
<b>2.5 Planificación.....</b>	<b>45</b>
<b>2.5.1 Estimación de esfuerzos por historias de usuarios. ....</b>	<b>45</b>
<b>2.5.2 Plan de duración de las iteraciones.....</b>	<b>47</b>
<b>2.5.3 Plan de entrega. ....</b>	<b>48</b>
<b>2.6 Iteraciones.....</b>	<b>49</b>
<b>2.7 Producción. ....</b>	<b>50</b>
<b>2.7.1 Diseño.....</b>	<b>50</b>
<b>2.7.2. Codificación. ....</b>	<b>54</b>
<b>CONCLUSIONES GENERALES.....</b>	<b>56</b>

## ÍNDICE DE FIGURAS

Figura 1: Pantalla de la aplicación.....	19
Figura 2: Pantalla de la aplicación.....	20
Figura 3: Representación del patrón de arquitectura Modelo-Vista-Controlador .....	52

## ÍNDICE DE TABLAS

Tabla 1: Tipo de matrices.....	22
Tabla 2.1 Personas relacionadas con el sistema.....	41
Tabla 2.2 Historias de usuario "Sumar matrices".....	42
Tabla 2.3 Historias de usuario " Visualizar suma de matrices".....	43
Tabla 2.4 Estimación de esfuerzos por historias de usuarios.....	45
Tabla 2.5 Plan de duración de las iteraciones.....	47
Tabla 2.6 Plan de entregas.....	48
Tabla 2.7 Tarea Mostrar formulario para sumar matrices.....	49
Tabla 2.8 Tarea Visualizar suma de matrices.....	50

## INTRODUCCIÓN

Desde el surgimiento de las computadoras, éstas se han presentado como un elemento útil y necesario de la vida cotidiana. Su utilización es fácilmente observable en profesiones tan diversas como la medicina, la educación, la ingeniería, la arquitectura, la administración, o en sectores como el gobierno, la industria, la banca o el comercio, lo cual se debe a que estos novedosos artículos tienen como virtud principal procesar con mucha facilidad y a gran velocidad enormes volúmenes de información.

La educación es uno de los campos donde han incursionado positivamente las Tecnologías de Información y las Comunicaciones (TIC), elemento clave en la construcción de la sociedad basada en la información, el conocimiento y el aprendizaje.

Parte considerable del desnivel entre personas, organizaciones, regiones y países se debe a la desigualdad de oportunidades relativas al desarrollo de la capacidad de aprender. Educar, en la sociedad de la información, significa más que capacitar a las personas para el uso de las TIC, consiste en invertir en la creación de competencias suficientemente amplias, que les permitan tener una actuación efectiva en el uso de la información para la producción de bienes y servicios, toma de decisiones y la creación de conocimientos. "Se trata de formar a los individuos para 'aprender a aprender' de modo que sean capaces de lidiar positivamente con la continua y acelerada transformación de la base tecnológica, mediante procesos de formación a lo largo de toda la vida" (Bonilla, Mendoza, & Lavidalie, 2005).

La integración de las TIC en el Proceso de Enseñanza-Aprendizaje (PEA) crea ambientes innovadores de aprendizaje, lo que permite el desarrollo de modelos y metodologías didácticas, de prototipos, materiales didácticos y la formación de comunidades académicas. Lo antes expuesto provoca la modernización de la práctica docente y la creación de ambientes virtuales de aprendizaje; elevando así el trabajo colaborativo. "De ahí que solo con la tecnología no basta, es importante adiestrar a los docentes para que después ellos propongan y desarrollen nuevas estrategias didácticas, cambiando su rol, para así incorporar plenamente y con ventajas las TIC" (Ortega, 2003).

Las TIC como herramientas de apoyo en la adquisición del conocimiento permiten:

- Educación sincrónica y asincrónica.
- Compilación, análisis y procesamiento de información.
- Favorecen el trabajo cooperativo.
- El uso eficiente y constante de los recursos de cómputo e informático.
- Traen el mundo al salón de clases y lleva el aula al ámbito global.
- Nuevos esquemas de gestión de conocimiento.
- Soporte para implementar nuevos métodos de enseñanza-aprendizaje e investigación.

Las TIC desde la perspectiva del aprendizaje elevan el interés y la motivación, se convierten en uno de los motores del aprendizaje, pues incitan a la actividad y al pensamiento. Por otro lado, la motivación hace que los estudiantes dediquen más tiempo a trabajar y, por tanto, aprenden más y mantienen una interacción continua (Aliaga, 2012).

La universidad cubana de hoy no se estructura solo a partir de determinadas demandas de tipo profesional, como ocurre en otros países, con un enfoque centrado en brindar rápida respuesta a las exigencias del mercado del trabajo, lanzando así a sus egresados a una competencia brutal por su subsistencia. No es esa la realidad cubana de hoy, y eso permite proyectar un modelo alternativo, con una mejor respuesta a las necesidades actuales del desarrollo económico y social. El paradigma está en brindar a la sociedad un profesional formado de manera íntegra, competente, con preparación científica para aceptar los retos de la sociedad moderna y con un amplio desarrollo humanístico para vivir en la sociedad de esta época y servirla con sencillez, modestia y otros valores como pilares fundamentales de su formación.

En Cuba se hace un gran esfuerzo para aprovechar los beneficios de esta tecnología fundamentalmente en la Educación Superior. Se han desarrollado varios software para apoyar el proceso enseñanza-aprendizaje de muchas materias (García, 2004), pero quedan algunas como el Álgebra Lineal, que aunque cuenta con buenos software estos no cumplen con todas las

expectativas pues se limitan solo a mostrar el resultado de las operaciones y no así los pasos que se siguen en cada una de ellas.

El departamento de Informática de la Universidad de Guantánamo, atiende la carrera de Ingeniería Informática. La misma tiene el propósito de formar profesionales con habilidades acorde a las necesidades productivas requeridas, con diversidad de perfiles, amplia flexibilidad curricular, alto nivel de creatividad; pues son el soporte de la informatización de la provincia.

Durante la carrera el estudiante recibe una serie de materias esenciales que contribuyen al desarrollo de las habilidades básicas que todo profesional de estos tiempos debe tener, para estar acorde con lo que demanda la sociedad actual. En el primer año de la carrera, los estudiantes reciben un conjunto de asignaturas fundamentales que sientan las bases en el ciclo de formación del futuro profesional, entre las que se encuentra el Álgebra Lineal.

Desde el comienzo de la carrera se imparte la asignatura y forma parte del ciclo básico. Durante los cursos que se ha impartido, ha sufrido disímiles cambios en los temas que dentro de ella se imparten, al constatar la carencia en las habilidades y conocimientos que el estudiante debería tener al culminar el año.

Se detectaron algunas insuficiencias por parte de los estudiantes en ciertos temas y específicamente en sistemas de ecuaciones lineales y matrices. En un estudio exploratorio realizado a una muestra de los estudiantes de la carrera en el tema de sistemas de ecuaciones lineales y matrices de la asignatura, se pudo observar que los alumnos tenían limitaciones en:

- Efectuar operaciones con matrices (suma, resta, multiplicación y multiplicación de un escalar por una matriz).
- Resolver sistemas de ecuaciones lineales utilizando la matriz ampliada del sistema y el procedimiento de eliminación de Gauss (estrategia de pivote elemental o parcial).
- Hallar el matriz escalón.
- Calcular el determinante mediante métodos exactos y métodos numéricos.
- Determinar la matriz traspuesta de una matriz.

- Cálculo de la matriz inversa.
- Verificar si una matriz es simétrica o no.

Esto se debe a que no dominan el contenido impartido y a la falta de una herramienta que les permitiera visualizar paso a paso la resolución de estas operaciones.

Por todo lo expuesto anteriormente se plantea como **problema a resolver**: ¿Cómo contribuir al aprendizaje del trabajo con matrices en la asignatura Álgebra Lineal de la carrera de Ingeniería Informática en la Universidad de Guantánamo?

El problema planteado está enmarcado en el **objeto de estudio**: El proceso de enseñanza-aprendizaje de la asignatura Álgebra Lineal en la carrera de Ingeniería Informática de la Universidad de Guantánamo.

Para contribuir a resolver el problema y a transformar el objeto de la investigación se propone como **objetivo**: Diseñar un sistema informático para el aprendizaje del trabajo con matrices en la asignatura Álgebra Lineal de la carrera de Ingeniería Informática en la Universidad de Guantánamo.

Lo que permite precisar como **campo de acción**: El aprendizaje del trabajo con matrices de la asignatura Álgebra Lineal en la carrera de Ingeniería Informática de la Universidad de Guantánamo.

En esta investigación se concibe como **idea a defender** que con el diseño de un sistema informático se contribuye a mejorar el aprendizaje del trabajo con matrices en la asignatura Álgebra Lineal de la carrera de Ingeniería Informática en la Universidad de Guantánamo.

Para dar cumplimiento al objetivo, se propusieron las siguientes **tareas**:

1. Caracterizar el proceso de enseñanza-aprendizaje de la asignatura Álgebra Lineal.
2. Fundamentar teóricamente los sistemas informáticos para el aprendizaje de trabajo con matrices.
3. Diagnosticar el estado actual del aprendizaje del trabajo con matrices en la asignatura Álgebra Lineal de la carrera de Ingeniería Informática.

4. Diseñar el sistema informático para el aprendizaje del trabajo con matrices en la asignatura Álgebra Lineal de la carrera de Ingeniería Informática.

### **Métodos de investigación**

Los métodos que han de utilizarse en la investigación y su selección, se relacionan estrechamente con la delimitación del objeto que se estudia, la determinación del objetivo y las tareas, además de dar la posibilidad de entender un conjunto de datos importantes que conducen a las conclusiones de la investigación.

A continuación se mencionan los métodos utilizados en las diferentes etapas de la investigación (León & González, 2002) y que revelan las características del objeto.

Como métodos en el nivel teórico se utilizaron:

- Análisis-Síntesis: se emplea para el análisis del Proceso de Enseñanza-Aprendizaje de la asignatura Álgebra Lineal, así como los sistemas informáticos para el aprendizaje del trabajo con matrices y determinar las características fundamentales de los mismos.
- Histórico-Lógico: en la determinación de las etapas principales de desarrollo del proceso de enseñanza-aprendizaje de la asignatura Álgebra Lineal, para el establecimiento de regularidades y tendencias de dicho proceso.

Como métodos empíricos se utilizaron:

- Encuesta a profesores y estudiantes para el completamiento del diagnóstico inicial y final y validar la propuesta como solución al problema de investigación.
- La observación: para penetrar en las manifestaciones fácticas del objeto de investigación y en el campo de acción.

Como técnica de recopilación de información se utilizó la entrevista a los estudiantes y profesores de la carrera de Ingeniería Informática, con el objetivo de buscar información que permita diagnosticar el problema.

**El documento de esta investigación está estructurado en dos capítulos, bibliografía y anexos:**

Capítulo 1: Fundamentación teórica. Se tratarán temas relacionados con el diseño teórico; además de abordar sobre la metodología de desarrollo, lenguajes, herramientas y tecnologías a emplear en el desarrollo del sistema.

Capítulo 2: Se profundizará en el proceso de diseño del sistema, las fases de la metodología seleccionada las cuales son: exploración, planificación de la entrega, iteraciones, producción, mantenimiento y muerte del proyecto.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción al capítulo.**

En el presente capítulo se valoran las tendencias históricas de asignatura Álgebra Lineal en la carrera de Ingeniería Informática. Se realiza una caracterización del objeto de investigación que constituye parte del fundamento teórico que sustenta la tesis. Se abordan temas referentes a software utilizados dentro del proceso de enseñanza-aprendizaje del Álgebra Lineal y operaciones con matrices, se da una breve argumentación teórica de las herramientas que son utilizadas para el desarrollo del software, así como la metodología a utilizar.

### **1.2 Análisis y surgimiento del Álgebra Lineal.**

La evolución de la Matemática hasta la actualidad ha sido producto del trabajo de muchas civilizaciones en la antigüedad, personalidades e instituciones a lo largo de todo el desarrollo humano y de diversas regiones del mundo que con sus aciertos y contradicciones filosóficas no han obstaculizado dicho avance.

El estudio de esta materia surge de la necesidad de resolver problemas de números y del deseo del hombre de comprender el universo que habita. La Matemática se cataloga como ciencia porque tiene objeto, leyes, categorías y métodos, además de un carácter universal y dialéctico.

De acuerdo con lo que se conoce hoy, el primer ejemplo de utilización sistemática del método axiomático-deductivo lo constituyen los *Elementos* de Euclides de Alejandría. Con este tratado lo matemático adquiriría por primera vez carácter científico. Con él nacería la Matemática como ciencia y comenzaría su historia (Urbaneja, 2002).

Diversos autores coinciden en la importancia de que los individuos comprendan el papel que desempeñan las matemáticas en el mundo, alcancen razonamientos fundados y participen en ellas en función de las necesidades de su vida; se ha enfatizado la necesidad de superar el anumerismo (Paulos, 1998) a través de un óptimo desempeño del pensamiento matemático y el desarrollo una Cultura Matemática. Particularmente, invocamos la importancia del pensamiento algebraico por resultar trascendente a nuestros fines; este asunto aparece relacionado con la calidad de las prácticas de enseñanza, razón por la que un sólido pensamiento de este tipo del educador matemático

se percibe como un requisito necesario para el desarrollo de un proceso de enseñanza orientado a lograr un nivel de aprendizaje de la Matemática en el que se propenda su desarrollo, en consecuencia resulta clave el proceso de formación inicial de este docente.

Además, este pensamiento se considera importante pues permite: deducir lo general en lo particular, hacer y revertir operaciones, reconocer patrones, abordar procesos de modelización, etc.

Por tanto, se considera importante que el educador matemático posea un nivel satisfactorio de desarrollo de su pensamiento algebraico a fin de que pueda promoverlo en sus estudiantes.

El Álgebra Lineal es una rama de las matemáticas modernas que juega un papel central debido a que se encarga del estudio de conceptos tales como vectores, matrices, sistemas de ecuaciones lineales, espacios vectoriales y transformaciones lineales. En Álgebra Lineal, los conceptos son tan importantes como los cálculos, por lo que se convierte en un curso adecuado para introducir el pensamiento abstracto, debido a que una gran parte de su campo tiene una interpretación geométrica, que puede ayudar precisamente a visualizar esos conceptos.

### **1.2.1 El papel de la historia del Álgebra Lineal en los Procesos de Enseñanza-Aprendizaje.**

Una de las funciones de la universidad es la formación de habilidades, específicamente en la actitud reflexiva. Lo investigativo se potencia en la perspectiva histórica, ya no se observa el conocimiento como un término acabado e inmutable, sino dinámico. Otro elemento a considerar es la erradicación de concepciones erróneas y expresiones ambiguas. Cuando por ejemplo se hace alusión a la Matemática Moderna, se hace necesario un recorrido histórico que permita revisar las connotaciones del término moderno, su aparición, propósitos, sentido y no su simple uso, pretendiendo hacer referencia a algo muy actual (Lozano, 2009).

La Matemática es una ciencia y atendiendo al concepto dado por Jorge Núñez Jover, ciencia es "Sistema de Conocimientos que modifica nuestra visión del mundo real y enriquece nuestra imaginación y nuestra cultura, se le puede comprender como proceso de investigación que permite obtener nuevos

conocimientos, los que a su vez ofrecen mayores posibilidades de manipulación de los fenómenos, es posible atender a sus impactos prácticos y productivos, caracterizándola como fuerza productiva que propicia la transformación del mundo y es fuente de riqueza, la ciencia también se nos presenta como una profesión debidamente institucionalizada portadora de su propia cultura y como funciones sociales bien identificadas". (Núñez, 1999).

Las formas y vías del desarrollo de los conocimientos matemáticos en los diferentes pueblos son muy diversas. Sin embargo, a pesar de las diferentes vías de desarrollo, es común para todos los pueblos que todos los conceptos básicos de las matemáticas surgieron de la práctica y atravesaron un largo período de perfeccionamiento. (Mola, 2002).

Uno de los conocimientos fundamentales de la matemática lo constituye la rama del Álgebra Lineal. Está diseñada para el logro de competencias específicas: números complejos, matrices, determinantes, sistema de ecuaciones lineales, espacios vectoriales, base y dimensión de un espacio vectorial y transformaciones lineales.

La "algebraización" de las Matemáticas, es decir, la influencia de las ideas y los métodos del álgebra en las Matemáticas son algunas de las características que configuran la matemática actual. Esta moda algebraica es un hecho fácilmente observable si se comparan textos de Matemáticas de años anteriores a la segunda guerra mundial, con los textos que aparecen a partir de los años sesenta. Aunque para muchos autores esta envoltura algebraica tiene un grave inconveniente cuando se usa en exceso (sobrecarga del formalismo que oscurece el contenido), sin embargo, el álgebra es desde la antigüedad una de las partes esenciales de las Matemáticas, al igual que lo es la geometría: "el álgebra no es otra cosa que la geometría escrita en símbolos, y la geometría es sencillamente álgebra expresada en figuras" (Sofía Germain, siglo XIX). Además, lo que sí parece ser aceptado universalmente es que la naturaleza de los objetos matemáticos es un hecho secundario, no importa mucho la forma en se presenten los resultados matemáticos, ya sea como un teorema de geometría pura o en forma de teorema algebraico: lo importante son las relaciones. Por ello el álgebra se ha definido como la ciencia de las operaciones algebraicas, efectuadas sobre los elementos de los distintos

conjuntos utilizados en Matemáticas. Una de las ramas más importantes del álgebra moderna y que más aplicaciones encuentra en otras parcelas de la ciencia como la física, la estadística, la ingeniería, el análisis numérico, las ciencias sociales, es el Álgebra Lineal (Guzmán, 1994).

La enseñanza-aprendizaje del Álgebra Lineal está frecuentemente asociada con procesos de aprendizaje memorístico que ofrecen una pobre visión de estos contenidos, lo cual obliga a realizar una fuerte revisión de su concepción, encaminada a ofrecer una visión más global y más profunda de la misma. Además, es muy frecuente que los conceptos del Álgebra Lineal se adquieran como formas sin contenido, es decir, un conjunto de relaciones simbólicas vacías de significado. De ahí, que los problemas algebraicos a veces carezcan del sentido necesario para la adquisición de un aprendizaje significativo.

El Álgebra Lineal es uno de los primeros contextos matemáticos con los cuales se enfrenta cualquier alumno (en algunos casos es la única experiencia matemática de toda su vida), circunstancia que configura esta parte de las Matemáticas como uno de los elementos y pilares básicos en la formación matemática de todo individuo.

Una de las asignaturas básicas de la mayor parte de las Ingenierías, suele estar basada en contenidos de Álgebra Lineal. Efectivamente, los contenidos básicos del Álgebra Lineal son el cimiento de una buena formación matemática para cualquier ingeniero o científico.

Hoy en día el Álgebra Lineal está presente en numerosos estudios universitarios, debido fundamentalmente al aumento general de las aplicaciones Matemáticas en áreas que por tradición no son de tipo técnico y por las aplicaciones originadas por la aparición de los computadores de alta velocidad. Esta influencia del Álgebra Lineal en otros campos de estudio ha obligado a que sea una de las materias presente en los programas de numerosos estudios universitarios.

### **1.3 Caracterización del proceso de enseñanza-aprendizaje de la asignatura Álgebra Lineal en la carrera de Ingeniería Informática.**

El Álgebra Lineal en la actualidad es uno de los temas centrales en el currículo de las ingenierías, no obstante, no se le ha dado en éste el lugar que el tema

se merece. La misma aporta al perfil del ingeniero, la capacidad de desarrollar un pensamiento lógico y algorítmico al resolver problemas. En las carreras de ingeniería tiene un papel muy importante ya que contribuye a desarrollar el pensamiento lógico, además de desarrollar la capacidad de preparar y tomar decisiones tecnológicas como parte del modo de actuación del ingeniero.

Desde que abriera sus puertas la carrera de Ingeniería Informática en la Universidad de Guantánamo, dentro de su plan de estudio estuvo contemplada la asignatura Álgebra Lineal perteneciente a la disciplina de Matemática, la cual estaría constituida por distintas asignaturas que formarían parte del ciclo básico de todo el profesional.

El objetivo de la disciplina de Matemática siempre ha sido el mismo: desarrollar las capacidades intelectuales para la modelación y creación de algoritmos, así como la formación computacional; además de que el estudiante se apropie de una serie de conocimientos y habilidades que deben tener una vez terminado el ciclo básico de la carrera, los cuales le servirán para asignaturas del ciclo profesional y en su desempeño en la actividad productiva.

Para realizar el análisis histórico tendencial de la asignatura Álgebra Lineal, el autor tiene en cuenta tres indicadores fundamentales, los cuales serán analizados por etapas respectivamente.

### **Etapa 2003-2008**

Indicador: Coherencia sistémica del contenido de la asignatura.

- Limitaciones en la integración de los contenidos de la asignatura.
- En poco se consideran los problemas de la profesión al modelar los contenidos, ejercicios, evaluaciones y problemas didácticos de la asignatura.
- Existe una carencia existencial de softwares educativos para la asignatura.

La característica principal de la etapa anterior son las limitaciones en el carácter sistémico y en la lógica de los contenidos de la asignatura de Álgebra Lineal.

### **Etapa 2008-2013**

Indicador: Vínculos de los contenidos de la asignatura con los problemas de la profesión.

- Aunque mejora la lógica de integración entre los contenidos aún persisten ligera secuenciación de estos.
- Persisten limitaciones en la problematización del contenido de la asignatura con arreglo a los problemas de la profesión.
- Se revelan las intenciones de insertar y diseñar Software Educativo para la asignatura, política que queda en condición de proyecto sin concretar su realización. Aunque los docentes comienzan a profundizar en la cultura en torno al tema.

Se revela como síntesis, en la etapa anterior, las deficiencias en la contextualización del contenido de Álgebra Lineal al perfil del profesional en formación.

### **Etapa 2013-actual**

Indicador: Pertinencia y estructura de los softwares educativos en la solución del problema didáctico.

- El trabajo metodológico realizado hace que la coherencia del contenido transite a niveles superiores de calidad, lo que se refleja en el diseño curricular y en las prácticas docentes de la asignatura.
- Se comienzan a revelar evidencias entre los problemas de la profesión y los problemas modelados desde el contenido de la asignatura.
- Se emplean aplicaciones de propósito específico de la Matemática como ciencia, que hace que sus contenidos sean generales con respecto al programa de la asignatura y su orden guarda poca relación con la lógica de las habilidades de la ciencia.

Se evidencia como aspecto generalizador de este estudio, que existe una tendencia al perfeccionamiento de los medios didácticos empleados en la asignatura de Álgebra Lineal, la cual está caracterizada por la inclusión de las TIC, aunque prevalece el carácter asistémico de estos.

#### **1.4 Software Educativo.**

Sánchez, en su Libro "Construyendo y Aprendiendo con el Computador", define el concepto genérico de software educativo como cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. Un concepto más restringido de software educativo lo define como aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender (Sánchez, 1999).

Según (Lamas & García, 2000), es una aplicación informática, que soportada sobre una bien definida estrategia pedagógica, apoya directamente el proceso de enseñanza-aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo.

Finalmente, un software educativo es un programa o aplicación realizada con la finalidad de ser utilizados como facilitadores del proceso de enseñanza.

Dentro de sus características generales, se puede encontrar que:

- Son materiales elaborados con una finalidad didáctica.
- Utilizan el computador como soporte en el que los alumnos realizan las actividades que ellos proponen.
- Son interactivos, "reaccionan" inmediatamente a las acciones de los estudiantes y permiten un diálogo y un intercambio de información entre el computador (o dispositivo) y los estudiantes.
- Individualizan el trabajo de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.
- Son fáciles de usar. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, aunque cada programa tiene reglas de funcionamiento que es necesario conocer.
- Reduce el tiempo de que se dispone para impartir gran cantidad de conocimientos, facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.

El uso de los softwares educativos en el proceso de enseñanza-aprendizaje puede ser:

### **Por parte del alumno**

Se evidencia cuando el estudiante opera directamente el software educativo, pero en este caso es de vital importancia la acción dirigida por el profesor.

### **Por parte del profesor**

Se manifiesta cuando el profesor opera directamente con el software y el estudiante actúa como receptor del sistema de información. La generalidad plantea que este no es el caso más productivo para el aprendizaje.

El uso del software por parte del docente proporciona numerosas ventajas, entre ellas:

- Enriquece el campo de la pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza-aprendizaje.
- Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.
- Pueden adaptar el software a las características y necesidades de su grupo teniendo en cuenta el diagnóstico en el proceso de enseñanza-aprendizaje.
- Permiten elevar la calidad del proceso docente-educativo.
- Permiten controlar las tareas docentes de forma individual o colectiva.
- Muestran la interdisciplinariedad de las asignaturas.
- Marca las posibilidades para una nueva clase más desarrolladora.

#### **1.4.1 Tipología del software educativo.**

Según Marqués, existen varias tipologías de software educativo y estas clasificaciones se basan en aspectos relacionados con los medios, las actividades cognitivas, las bases psicopedagógicas, etc. (Marqués, 1999).

Según los **objetivos educativos**: pretenden facilitar el aprendizaje de conceptos, procedimientos y/o actitudes.

Según los **medios** que integra: convencionales, los que incorporan multimedia, aquellos que están basados en el paradigma hipertexto y los que hacen uso de la realidad virtual.

Según las **actividades cognitivas** que activa: pueden promover la observación, la memorización, el control psicomotriz, la comprensión, la interpretación, la comparación, el análisis, síntesis, basados en resolución de problemas, como medio de expresión, para la creación y la experimentación.

Según las **bases psicopedagógicas** sobre el aprendizaje: basados en el conductismo, el cognitivismo o el constructivismo.

Según la **función** en la **estrategia didáctica**: se utilizan para entrenar, para instruir, informar, motivar, explorar, experimentar, evaluar, entretener, etc.

Según el **diseño**: centrado en el aprendizaje, centrado en la enseñanza, centrado en el estudiante, proveedor de recursos, entre otros.

### **Clasificaciones de software educativo**

- Prácticas.
- Tutoriales.
- Simulaciones.
- Juegos instruccionales.

### **Estructura de las diferentes clasificaciones de software educativo**

#### **Prácticas**

- Presentación de la introducción.
- Selección de ítems.
- Respuestas.
- Evaluación de respuestas.
- Retroalimentación.

#### **Tutoriales**

- Introducción.

- Presentación de la información.
- Formulación de preguntas interactivas.
- Respuestas.
- Evaluación de respuestas.
- Retroalimentación.

### **Simulaciones**

- Introducción.
- Presentación del fenómeno.
- Acción requerida.
- Acción del aprendiz.
- Alteración del sistema.

### **Juegos instruccionales**

- Introducción.
- Presentación del objetivo.
- Acción requerida.
- Acción del aprendiz y/o oponente.
- Alteración del sistema.

**Tutoriales directivos:** presentan información, hacen preguntas a los estudiantes y controlan en todo momento las actividades.

**Tutoriales no directivos:** el computador adopta el papel de un laboratorio o instrumento. El estudiante tiene “libertad” de acción.

**Tutoriales, y tutoriales de ejercitación:** dirigen las actividades del estudiante.

**Bases de datos:** proporcionan un entorno estático y facilitan la exploración y consulta selectiva.

**Simuladores:** proporcionan un entorno dinámico y facilitan la exploración y la manipulación del entorno.

**Constructores:** son entornos programables. Facilitan a los usuarios unos elementos simples con los cuales pueden construir elementos más complejos u otros entornos.

**Programas herramientas:** son programas que proporcionan un entorno instrumental con el cual se facilita la realización de ciertos trabajos generales de tratamiento de la información: escribir, organizar, calcular, dibujar, transmitir, captar datos.

El autor propone en esta investigación un software educativo que tiene la siguiente tipología, pretende facilitar el aprendizaje de conceptos, procedimientos y/o actitudes, que permita promover la comprensión, la interpretación, la comparación y el análisis; basado en la resolución de ejercicios, como medio de expresión para la creación, además de que se utilice para entrenar, instruir, informar, motivar, explorar, experimentar y entretener; centrado en el aprendizaje y en los estudiantes. El software propuesto se clasifica como un tutorial.

La clasificación está sustentada porque el software desarrollado presenta la siguiente estructura:

- Introducción.
- Presentación de la información.
- Respuestas.
- Visualización.
- Retroalimentación.

### **1.5 Sistemas informáticos similares.**

Durante el desarrollo de la investigación se realizó una búsqueda a nivel internacional y nacional sobre sistemas informáticos existente para el trabajo con matrices. A continuación, se presentan los principales resultados obtenidos.

#### **A nivel internacional**

**MATLAB:** Es un programa interactivo que ayuda a realizar cálculos numéricos, analizando y visualizando los datos, para resolver problemas matemáticos,

físicos, etc. Trabaja con escalares, vectores y matrices. Es un medio computacional técnico, con un gran desempeño para el cálculo numérico computacional y de visualización. Integra análisis numérico, matrices, procesamiento de señales y gráficas, todo esto en un ambiente donde los problemas y soluciones son expresados tal como se escriben matemáticamente.

Permite realizar de una manera simultánea una gran variedad de operaciones matemáticas, además de poderse trabajar con distintas plataformas según la potencia del software y del hardware disponible. Es un software propietario y fue desarrollado en C/C++ (García, 2004).

**DERIVE 6:** es un programa de álgebra computacional desarrollado en el mercado en 1988. Debido al poco consumo de memoria que requiere, es conveniente para ser usado en ordenadores antiguos o equipos con poca potencia, como calculadoras Texas Instrument (TI). Se encuentra disponible para las plataformas Windows y DOS, y es usado ampliamente con propósitos educativos. Es un asistente matemático con las siguientes posibilidades: Aritmética, Álgebra, Gráficos 2D y 3D, Cálculo, Vectores y Matrices, Funciones y Programación. Derive es muy sencillo de usar, y cuenta con una interfaz atractiva e intuitiva. Permite sumar, multiplicar, transponer e invertir matrices (García, 2004).

**MatrizAPP 1.0:** es una aplicación, gratis y multiplataforma disponible tanto en Windows como en Android. Este programa permite realizar cálculos para las siguientes operaciones (García, 2004):

- Suma y Resta de dos matrices.
- Producto de una matriz por un escalar.
- Producto de dos matrices.
- Matriz opuesta y transpuesta.

A continuación, se visualiza una imagen con una de las pantallas de la aplicación:

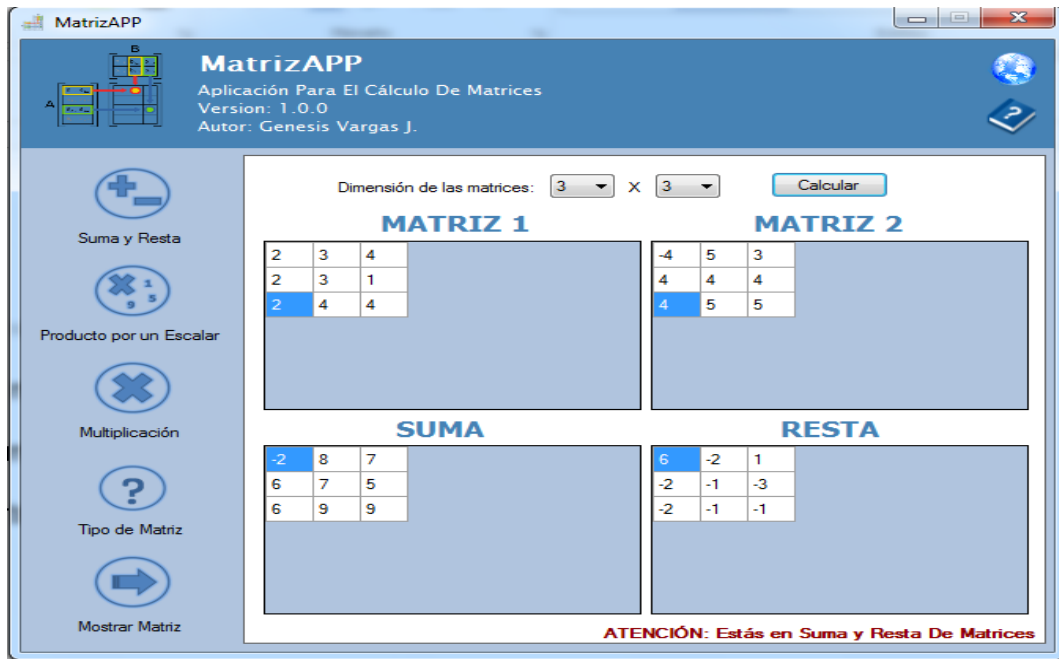


Figura 1: Pantalla de la aplicación. Suma y resta de matrices.

## A nivel nacional

**Matriz:** es un software educativo desarrollado en la carrera de Ingeniería Informática de la Universidad de Granma (UDG). Este software es una aplicación de escritorio y fue desarrollado en C++. Tiene un ambiente gráfico amigable, una buena precisión de los cálculos y que, principalmente, visualiza de forma dinámica y sencilla todas las operaciones matriciales (Almaguel, 2008).

A continuación, se visualiza una imagen con una de las pantallas de la aplicación:



Figura 2: Pantalla de la aplicación. Descripción de las opciones principales.

Luego de un análisis de los sistemas informáticos similares estudiados, tanto a nivel nacional como internacional, se decidió realizar una aplicación web. Además, se demuestra que estos sistemas no se pueden utilizar por las siguientes razones:

- Los sistemas internacionales realizan las operaciones básicas, sin embargo, no permiten visualizar las operaciones paso a paso.
- Los sistemas nacionales e internacionales estudiados no permiten una correcta portabilidad de la aplicación.
- Estos sistemas no cumplen con el diseño responsivo, característica indispensable en la actualidad teniendo en cuenta la variedad de dispositivos que existen para visualizar las distintas aplicaciones.
- No se ajustan a los requerimientos del cliente.

Fueron identificadas una serie de funcionalidades que son comunes en varias de estas aplicaciones y que pueden ser implementadas en el sistema que se pretende desarrollar, como, por ejemplo: visualización de algunas operaciones, la estructuración del contenido y el diseño de las principales operaciones básicas.

## 1.6 Matrices.

Para un mejor entendimiento del tema a tratar en el sistema informático a desarrollar, es preciso destacar los conceptos generales y la metodología que se sigue para resolver operaciones con matrices.

Las matrices aparecen por primera vez hacia el año 1850, introducidas por Silvestre. El desarrollo inicial de la teoría se debe al matemático Hamilton en 1853. En 1858, Cayley introduce la notación matricial como una forma abreviada de escribir un sistema de  $m$  ecuaciones lineales con  $n$  incógnitas.

Las matrices se utilizan en el cálculo numérico, en la resolución de sistemas de ecuaciones lineales, de las ecuaciones diferenciales y de las derivadas parciales. Además de su utilidad para el estudio de sistemas de ecuaciones lineales, las matrices aparecen de forma natural en geometría, estadística, economía, informática, física, etc.

La utilización de matrices constituye actualmente una parte esencial de los lenguajes de programación, ya que la mayoría de los datos se introducen en los ordenadores como tablas organizadas en filas y columnas: hojas de cálculo, bases de datos, etc.

### 1.6.1 Concepto de matriz.

Una matriz es un conjunto de elementos de cualquier naturaleza, aunque, en general, suelen ser números ordenados en filas y columnas.

Se llama matriz de orden " $m \times n$ " a un conjunto rectangular de elementos  $a_{ij}$  dispuestos en  $m$  filas y en  $n$  columnas. El orden de una matriz también se denomina dimensión o tamaño, siendo  $m$  y  $n$  números naturales (Barbolla, 1999).

Las matrices se denotan con letras mayúsculas:  $A, B, C, \dots$  y los elementos de las mismas con letras minúsculas y subíndices que indican el lugar ocupado:  $a, b, c, \dots$ . Un elemento genérico que ocupe la fila  $i$  y la columna  $j$  se escribe  $a_{ij}$ . Si el elemento genérico aparece entre paréntesis también representa a toda la matriz:

$$A = (a_{ij}).$$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & a_{ij} & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Ej.  $A_{2 \times 3} = \begin{pmatrix} \frac{1}{6} & -3 & 5 \\ 7 & \sqrt{2} & 4 \end{pmatrix}$

donde sus filas son :  $\left(\frac{1}{6} \quad -3 \quad 5\right)$  y  $\left(7 \quad \sqrt{2} \quad 4\right)$

y sus columnas  $\begin{pmatrix} \frac{1}{6} \\ 7 \end{pmatrix}$ ,  $\begin{pmatrix} -3 \\ \sqrt{2} \end{pmatrix}$  y  $\begin{pmatrix} 5 \\ 4 \end{pmatrix}$

- Cuando se refiere indistintamente a filas o columnas se habla de líneas.
- El número total de elementos de un matriz  $A_{m \times n}$  es  $m \cdot n$
- En matemáticas, tanto las listas como las tablas reciben el nombre genérico de matrices.

### 1.6.2 Tipos de matrices.

Hay algunas matrices que aparecen frecuentemente y que, según su forma, reciben nombres diferentes:

Tabla 1: Tipo de matrices.

Tipo de matriz	Definición	Ejemplo
FILA	Aquella matriz que tiene una sola fila, siendo su orden $1 \times n$	$A_{1 \times 3} = (7 \quad 2 \quad -5)$
COLUMNA	Aquella matriz que tiene una sola columna, siendo su orden $m \times 1$	$A_{3 \times 1} = \begin{pmatrix} -7 \\ 1 \\ 6 \end{pmatrix}$
RECTANGULAR	Aquella matriz que tiene distinto número de filas que de columnas, siendo su orden $m \times n$ , $m \neq n$	$A_{3 \times 4} = \begin{pmatrix} 1 & 3 & 2 & 9 \\ 5 & 7 & -1 & 8 \\ 0 & 3 & 5 & 1 \end{pmatrix}$

<p>TRASPUESTA</p>	<p>Dada una matriz A, se llama traspuesta de A a la matriz que se obtiene cambiando ordenadamente las filas por las columnas. Se representa por <math>A^t</math> ó <math>A^T</math></p>	<p>Si es <math>A = (a_{ij})_{m \times n}</math>  su traspuesta es <math>A^t = (a_{ji})_{n \times m}</math></p> $A = \begin{pmatrix} 1 & 2 & 5 \\ 3 & -4 & 7 \end{pmatrix}; A^t = \begin{pmatrix} 1 & 3 \\ 2 & -4 \\ 5 & 7 \end{pmatrix}$
<p>OPUESTA</p>	<p>La matriz opuesta de una dada es la que resulta de sustituir cada elemento por su opuesto. La opuesta de A es -A.</p>	$A = \begin{pmatrix} 1 & 3 \\ 5 & -7 \\ -6 & 4 \end{pmatrix}, -A = \begin{pmatrix} -1 & -3 \\ -5 & 7 \\ 6 & -4 \end{pmatrix}$
<p>NULA</p>	<p>Si todos sus elementos son cero. También se denomina matriz cero y se denota por <math>0_{m \times n}</math></p>	$0_{3 \times 4} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
<p>CUADRADA</p>	<p>Aquella matriz que tiene igual número de filas que de columnas, <math>m = n</math>, diciéndose que la matriz es de orden n.  <u>Diagonal principal</u> : son los elementos <math>a_{11}</math> , <math>a_{22}</math> , ..., <math>a_{nn}</math>  <u>Diagonal secundaria</u> : son los elementos <math>a_{ij}</math> con <math>i+j = n+1</math>  <u>Traza</u> de una matriz cuadrada : es la suma de los elementos de la diagonal principal de A.</p>	$A_3 = \begin{pmatrix} 1 & 9 & -6 \\ 0 & 2 & 1 \\ -2 & 4 & 5 \end{pmatrix}$ <p>Diagonal principal :</p> $A = \begin{pmatrix} 5 & 1 & 5 & -6 \\ 1 & 2 & 0 & 3 \\ 7 & -3 & 4 & 11 \\ 1 & 9 & 3 & 8 \end{pmatrix}$ <p>Diagonal secundaria :</p> $A = \begin{pmatrix} 5 & 1 & 5 & -6 \\ 1 & 2 & 0 & 3 \\ 7 & -3 & 4 & 11 \\ 1 & 9 & 3 & 8 \end{pmatrix}$

SIMÉTRICA	<p>Es una matriz cuadrada que es igual a su traspuesta.</p> <p><math>A = A^t</math> , <math>a_{ij} = a_{ji}</math></p>	$A_3 = \begin{pmatrix} 1 & 9 & -6 \\ 9 & 2 & 1 \\ -6 & 1 & 5 \end{pmatrix}$
ANTISIMÉTRICA	<p>Es una matriz cuadrada que es igual a la opuesta de su traspuesta.</p> <p><math>A = -A^t</math> , <math>a_{ij} = -a_{ji}</math></p> <p>Necesariamente <math>a_{ii} = 0</math></p>	$A_3 = \begin{pmatrix} 0 & 9 & -6 \\ 9 & 0 & 1 \\ -6 & 1 & 0 \end{pmatrix}$
DIAGONAL	<p>Es una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal</p>	$A = \begin{pmatrix} 7 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & -2 \end{pmatrix}$
ESCALAR	<p>Es una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal que son iguales</p>	$A = \begin{pmatrix} 7 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 7 \end{pmatrix}$
IDENTIDAD	<p>Es una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal que son iguales a 1. También se denomina matriz unidad.</p>	$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
TRIANGULAR	<p>Es una matriz cuadrada que tiene todos los elementos por encima (por debajo) de la diagonal principal nulos.</p>	$A = \begin{pmatrix} 1 & 3 & 5 \\ 0 & 4 & -1 \\ 0 & 0 & 9 \end{pmatrix} \quad T. \text{ superior}$ $A = \begin{pmatrix} 1 & 0 & 0 \\ 5 & 4 & 0 \\ 2 & 8 & 7 \end{pmatrix} \quad T. \text{ inferior}$

ORTOGONAL	<p>Una matriz ortogonal es necesariamente cuadrada e invertible: <math>A^{-1} = A^T</math></p> <p>La inversa de una matriz ortogonal es una matriz ortogonal.</p> <p>El producto de dos matrices ortogonales es una matriz ortogonal.</p> <p>El determinante de una matriz ortogonal vale +1 ó -1.</p>	$A \cdot A^T = A^T \cdot A = I$ $\begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \cdot \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$
NORMAL	<p>Una matriz es normal si conmuta con su traspuesta. Las matrices simétricas, antisimétricas u ortogonales son necesariamente normales.</p>	$A = \begin{pmatrix} 5 & 4 \\ -4 & 5 \end{pmatrix}$ $A \cdot A^T = A^T \cdot A$
INVERSA	<p>Una matriz cuadrada A tiene inversa, <math>A^{-1}</math>, si se verifica que:</p> $A \cdot A^{-1} = A^{-1} \cdot A = I$	$A = \begin{pmatrix} 2 & 3 \\ 1 & 1 \end{pmatrix}; A^{-1} = \begin{pmatrix} -1 & 3 \\ 1 & -2 \end{pmatrix}$

### 1.6.3 Operaciones con matrices.

#### ➤ Suma de matrices

La suma de dos matrices  $A = (a_{ij})_{m \times n}$  y  $B = (b_{ij})_{p \times q}$  de la misma dimensión (equidimensionales) :  $m = p$  y  $n = q$  es otra matriz  $C = A+B = (c_{ij})_{m \times n} = (a_{ij}+b_{ij})$  (Cayley, 1858).

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} ; \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

$$A + B = \begin{pmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \end{pmatrix}$$

Al conjunto de las matrices de dimensión  $m \times n$  cuyos elementos son números reales se va a representar por  $M_{m \times n}$ .

### ➤ Producto de un escalar por una matriz

Para multiplicar un escalar por una matriz se multiplica el escalar por todos los elementos de la matriz, obteniéndose otra matriz del mismo orden.

$$A = (a_{ij}) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} ; \quad \lambda \cdot A = \begin{pmatrix} \lambda a_{11} & \lambda a_{12} & \dots & \lambda a_{1n} \\ \lambda a_{21} & \lambda a_{22} & \dots & \lambda a_{2n} \\ \dots & \dots & \dots & \dots \\ \lambda a_{m1} & \lambda a_{m2} & \dots & \lambda a_{mn} \end{pmatrix}$$

*p. ejm.*

$$A = \begin{pmatrix} 1 & -2 & 3 \\ 0 & 1 & 8 \end{pmatrix} ; \quad (-5) \cdot A = \begin{pmatrix} -5 & 10 & -15 \\ 0 & -5 & -40 \end{pmatrix}$$

### ➤ Producto de matrices

Dadas dos matrices  $A = (a_{ij})_{m \times n}$  y  $B = (b_{ij})_{p \times q}$  donde  $n = p$ , es decir, el número de columnas de la primera matriz  $A$  es igual al número de filas de la matriz  $B$ , se define el producto  $A \times B$  de la siguiente forma:

El elemento que ocupa el lugar  $(i, j)$  en la matriz producto se obtiene sumando los productos de cada elemento de la fila  $i$  de la matriz  $A$  por el correspondiente de la columna  $j$  de la matriz  $B$  (Cayley, 1858).

### ➤ Matriz inversa

Se llama matriz inversa de una matriz cuadrada  $A_n$  y la representamos por  $A^{-1}$ , a la matriz que verifica la siguiente propiedad :  $A^{-1} \cdot A = A \cdot A^{-1} = I$  (Cayley, 1858). Una matriz cuadrada es "*regular*" si su determinante es distinto de cero, y es "*singular*" si su determinante es igual a cero.

$|A| \neq 0 \Rightarrow$  *Matriz Regular*

$|A| = 0 \Rightarrow$  *Matriz Singular*

- ✓ Sólo existe matriz inversa de una matriz cuadrada si ésta es *regular*.
- ✓ La matriz inversa de una matriz cuadrada, si existe, es única.
- ✓ Entre matrices NO existe la operación de división, la matriz inversa realiza funciones análogas.

### ➤ **Método de gauss**

El **Método de Gauss** es una generalización del método de reducción, que utilizamos para eliminar una incógnita en los sistemas de dos ecuaciones con dos incógnitas. Consiste en la aplicación sucesiva del método de reducción, utilizando los criterios de equivalencia de sistemas para transformar la matriz ampliada con los términos independientes ( $A^*$ ) en una *matriz triangular*, de modo que cada fila (ecuación) tenga una incógnita menos que la inmediatamente anterior. Se obtiene así un sistema, que llamaremos escalonado, tal que la última ecuación tiene una única incógnita, la penúltima dos incógnitas, la antepenúltima tres incógnitas, ..., y la primera todas las incógnitas (Guido, 1997).

El siguiente esquema muestra cómo se puede resolver un sistema de ecuaciones lineales aplicando este método.

Se parte, inicialmente, de un sistema de  $n$  ecuaciones lineales con  $n$  incógnitas, compatible determinado:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n = b_3$$

.....

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

En primer lugar, aplicando sucesivamente el método de reducción, se elimina en todas las ecuaciones, excepto en la primera, la incógnita  $x_1$ , obteniéndose un sistema equivalente:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$c_{22}x_2 + \dots + c_{2n}x_n = r_2$$

$$c_{32}x_2 + \dots + c_{3n}x_n = r_3$$

.....

$$c_{n2}x_2 + \dots + c_{nn}x_n = r_n$$

En segundo lugar, aplicando nuevamente el método de reducción de forma sucesiva, se elimina en todas las ecuaciones, excepto en las dos primeras, la incógnita  $x_2$ , obteniéndose un sistema equivalente:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$c_{22}x_2 + \dots + c_{2n}x_n = r_2$$

$$\dots + d_{3n}x_n = s_3$$

.....

$$\dots + d_{nn}x_n = s_n$$

En tercer lugar, aplicando sucesivamente el método de reducción, se elimina en todas las ecuaciones, excepto en las tres primeras, la incógnita (...), y así sucesivamente, hasta obtener el siguiente sistema equivalente:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$c_{22}x_2 + \dots + c_{2n}x_n = r_2$$

$$\dots + d_{3n}x_n = s_3$$

.....

$$e_{nn}x_n = t_n$$

Para resolverlo se despeja, en primer lugar, la única incógnita de la última ecuación. Luego se sustituye ésta en la penúltima ecuación y se despeja la otra incógnita. Posteriormente, se sustituye dos de las tres incógnitas de la antepenúltima ecuación por sus valores y se despeja la que queda, y así sucesivamente hasta llegar a la primera ecuación.

### ➤ **Determinante**

El *determinante* de una matriz  $\mathbf{A}(n,n)$ , es un escalar o polinomio, que resulta de obtener todos los productos posibles de una matriz de acuerdo a una serie de restricciones, siendo denotado como  $|\mathbf{A}|$ . El valor numérico es conocido también como módulo de la matriz.

**DEFINICIÓN.** El determinante de A es la suma de los  $n!$  productos que se pueden formar eligiendo un elemento por fila y uno por columna de A (dicha elección coincide con una matriz de permutación), cada producto está además multiplicado por el signo de la paridad del número de inversiones o intercambios necesarios para convertir la permutación indicada en la identidad (Salvador, 1998).

$$\det A = \sum_{\sigma} a_{1\sigma(1)} \cdots a_{n\sigma(n)} (-1)^{\text{sign}(\sigma)}$$

En el caso particular de matrices de  $2 \times 2$ ,

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$$

A continuación, se muestra una de las formas de obtener el determinante:

A cada matriz  $n$ -cuadrada  $A = (a_{ij})$  se le asigna un escalar particular denominado determinante de A, denotado por  $\det(A)$ ,  $|A|$  o

$$\begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{m1} & \cdots & a_{mn} \end{vmatrix}$$

Una tabla ordenada  $n \times n$  de escalares situada entre dos líneas verticales, llamada determinante de orden  $n$ , no es una matriz. La función determinante apareció por primera vez en el estudio de los sistemas de ecuaciones lineales.

### Determinantes de orden uno y dos

Los determinantes de orden uno y dos se definen como sigue:

$$|a_{11}| = a_{11} \quad \text{y} \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} * a_{22} - a_{12} a_{21}$$

Así, el determinante de una matriz  $1 \times 1$   $A = (a_{11})$  es el propio escalar  $a_{11}$ , es decir,  $\det(A) = |a_{11}| = a_{11}$ .

## Determinantes de orden tres

Considere una matriz 3 x 3 arbitraria  $A = (a_{ij})$ . El determinante de  $A$  se define como sigue:

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} * a_{22} * a_{33} + a_{12} * a_{23} * a_{31} + a_{21} * a_{32} * a_{13} - a_{13} * a_{22} * a_{31} - a_{12} * a_{21} * a_{33} - a_{32} * a_{23} * a_{11}$$

Obsérvese que hay seis productos, cada uno formado por tres elementos de la matriz. Tres de los productos aparecen con signo positivo (conservan su signo) y tres con signo negativo (cambian su signo).

Para calcular los determinantes de orden tres, los siguientes pasos pueden ayudar a resolverlos:

1- Se agregan la columna 1 y 2 al final. Como se muestra a continuación.

$$\begin{matrix} a_{11} & a_{12} & a_{13} & a_{11} & a_{12} \\ a_{21} & a_{22} & a_{23} & a_{21} & a_{22} \\ a_{31} & a_{32} & a_{33} & a_{31} & a_{32} \end{matrix}$$

2 - Se realiza el cálculo teniendo en cuenta que los productos positivos son los elementos de las diagonales principales (verde claro) y productos negativos son los elementos de las diagonales secundarias (verde fuerte).

El diagrama muestra una matriz 3x5 con los elementos  $a_{ij}$  en sus celdas. Se han trazado dos diagonales de color verde claro que conectan los elementos  $a_{11}, a_{22}, a_{33}$  y  $a_{12}, a_{23}, a_{31}$ . Se han trazado también dos diagonales de color verde fuerte que conectan los elementos  $a_{13}, a_{21}, a_{32}$  y  $a_{11}, a_{23}, a_{32}$ .

Cuando el determinante de una matriz resulta igual a 0 se dice que la matriz es no singular.

### 1.7 Metodología de desarrollo de software.

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema informático. Una gran variedad de metodologías se ha desarrollado a lo largo de los años, cada una de ellas con sus fortalezas y debilidades. Una

determinada metodología no es necesariamente aplicable a todo tipo de proyectos, más bien cada tipo de proyecto tiene una metodología a la que se adapta mejor (Barzaba, 2006).

Actualmente existen numerosas propuestas de metodologías que inciden en distintas dimensiones del proceso de desarrollo. Un ejemplo de ellas son las propuestas tradicionales centradas específicamente en el control del proceso. Estas han demostrado ser efectivas y necesarias en un gran número de proyectos, sobre todo aquellos proyectos de gran tamaño (respecto a tiempo y recursos). Sin embargo, la experiencia ha demostrado que las metodologías tradicionales no ofrecen una buena solución para proyectos donde el entorno es volátil y donde los requisitos no se conocen con exactitud. Como respuesta a los problemas aplicando metodologías tradicionales surgieron otras metodologías que tratan de adaptarse a la realidad del desarrollo de software, las metodologías ágiles. Existe una notable diferencia y es que las metodologías ágiles son menos orientadas al documento, exigiendo menos documentación para una determinada tarea dada (ISSI, 2003).

### **1.7.1 Extreme Programming.**

Extreme Programming (XP) es una metodología ágil, centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software. También se encarga de promover el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Además, se basa en la realimentación continua entre el cliente y el equipo de desarrollo, existiendo así una comunicación fluida entre todos los participantes. Se encuentra la simplicidad que está presente en las soluciones implementadas y el coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico (Joskowicz, 2008).

Su autor principal Kent Beck define un conjunto de cinco valores que establecen el fundamento para todo trabajo realizado en XP, tales como (Pressman, 2010):

- **Comunicación:** es muy importante que haya una comunicación constante con el cliente y dentro de todo el equipo de trabajo, de esto dependerá

que el desarrollo se lleve a cabo de una manera sencilla, entendible y que se entregue al cliente lo que necesita.

- Simplicidad: el diseño debe ser sencillo y amigable al usuario, el código debe ser simple y entendible, programando solo lo necesario y lo que se utilizará.
- Retroalimentación: es la comunicación constante entre el desarrollador y el usuario.
- Coraje: se refiere a la valentía que se debe tener al modificar o eliminar el código que se realizó con tanto esfuerzo; el desarrollador debe saber cuándo el código que desarrolló no es útil en el sistema y, por lo mismo, debe ser eliminado. También se refiere a tener la persistencia para resolver los errores en la programación.

La programación extrema cuenta con principios que guían el desarrollo de esta metodología. Algunos de estos principios son:

- El juego de la planificación.
- Entregas pequeñas.
- Metáfora.
- Diseño simple.
- Pruebas.
- Refactorización.
- Programación en parejas.
- Propiedad colectiva del código.
- Integración continua.
- La semana de 40 horas.
- Presencia del cliente.
- Estándar de codificación Ventajas

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.

Las metodologías ágiles ofrecen una solución casi a medida para una gran cantidad de proyectos que tienen estas características. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo (ISSI, 2003).

La metodología de desarrollo de software que se utilizará es XP, atendiendo principalmente a la simplicidad al desarrollar y codificar los módulos del sistema, además el principio de la refactorización permite remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios, así como la participación del cliente, quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio, permitiendo que se pueda resolver de manera inmediata cualquier duda asociada.

## **1.8 Lenguajes, herramientas y tecnologías para el desarrollo del sistema informático.**

En este epígrafe se realizó un estudio sobre los posibles lenguajes, herramientas y tecnologías a utilizar en el desarrollo de la aplicación, teniendo presente las características de cada una de ellas, así como sus tendencias actuales y novedades, con la intención de mantener la mayor integración permisible además de seleccionar la herramienta adecuada para la elaboración del programa informático que se propone como solución en esta investigación.

### **1.8.1 Aplicaciones de escritorio.**

Una aplicación de escritorio (también llamada Desktop) es aquella que está instalada en el ordenador del usuario, que es ejecutada directamente por el sistema operativo, y cuyo rendimiento depende de diversas configuraciones de hardware como memoria RAM (*Random-Access Memory* o memoria de acceso aleatorio), disco duro y memoria de video. Se utiliza como una herramienta para realizar una operación o tarea específica (Streetdirectory, 2015).

Algunas de las desventajas del uso de estas aplicaciones son las siguientes: su accesibilidad es solo en la computadora donde se haya instalado previamente el software, se requiere de instalación y actualización personalizada, es decir hay que realizarlos en cada estación de trabajo donde se tenga la aplicación y

además estas aplicaciones de escritorio generalmente se realizan para un sistema operativo específico (García & Gertrudix, 2011).

### **1.8.2 Aplicaciones web.**

Una aplicación web es un conjunto de páginas que interactúan unas con otras. Esta interacción permite implementar características en su sitio como catálogos de productos virtuales, administradores de noticias y contenidos. Adicionalmente podrá realizar consultas a bases de datos, registrar e ingresar información, solicitudes, pedidos y múltiples tipos de información en línea en tiempo real. Algunas de las ventajas de su uso sobre las aplicaciones de escritorio son las siguientes: se puede usar desde cualquier lugar, siempre y cuando exista una conexión a la red, no se requiere hacer actualizaciones o instalaciones en los clientes, y además, no se le obliga al usuario a usar un determinado sistema operativo (Rodríguez, Codina, & Pedraza, 2010).

Algunas de las ventajas de utilizar aplicaciones web son las siguientes (Magazine, 2015):

- Ahorran costes de hardware y software: solo es necesario usar un ordenador con un navegador web y conectarse a Internet. Las aplicaciones basadas en la web usan menos recursos que los programas instalados.
- Fáciles de usar: son muy sencillas de utilizar, solo necesitará conocimientos básicos de informática para trabajar con ellas.
- Facilitan el trabajo colaborativo y a distancia: pueden ser usadas por varios usuarios al mismo tiempo. Al estar toda la información centralizada no tendrá que compartir pantallas o enviar emails con documentos adjuntos. Varios usuarios pueden ver y editar el mismo documento de manera conjunta. Además, son accesibles desde cualquier lugar. Puede trabajar desde una PC, un portátil, un móvil o una Tablet, desde la oficina, parque o un aeropuerto.
- Provocan menos errores y problemas: son menos propensas a suspender y crear problemas técnicos debido a conflictos con hardware, con otras aplicaciones existentes, protocolos o con un software personal interno. Todos los usuarios utilizan la misma versión de la aplicación web y los posibles fallos pueden ser corregidos tan pronto como son descubiertos.

- Emplean tecnologías que permiten una gran portabilidad entre diferentes plataformas. Por ejemplo, una aplicación web podría ejecutarse en un dispositivo móvil, en una computadora con Windows, Linux u otro sistema, incluso en una consola de videojuegos, etc.

Para la investigación se selecciona una aplicación web, teniendo en cuenta algunas de las ventajas mencionadas anteriormente. En la actualidad las tecnologías web han evolucionado hasta el punto de que no existen diferencias notables con las aplicaciones de escritorio. Para el desarrollo del software se tendrá en cuenta las características que ofrecen estas tecnologías para el diseño y trabajo con los formularios. Se utilizará además el diseño responsivo uno de los estándares que en la actualidad deben cumplir las aplicaciones para lograr que se visualicen correctamente en la gran variedad de dispositivos que existen. Otras de las características a tener en cuenta es la portabilidad de la aplicación.

### **1.8.3 Lenguajes de programación web.**

En el presente epígrafe, se explica brevemente las tendencias y tecnologías actuales a utilizar para el desarrollo de la aplicación en cuestión. Actualmente existen diferentes lenguajes de programación para desarrollo web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Se clasifican como: del lado del cliente y del lado del servidor. Ambas clasificaciones son utilizadas para el desarrollo completo de una aplicación web. En el caso de la presente investigación se utilizarán para el desarrollo de la aplicación los lenguajes del lado del cliente.

Los sistemas basados en lenguajes de programación web crean cierto atractivo para un sin número de usuarios, quienes no dependerán por completo de un sistema operativo en particular para usar las aplicaciones, pues estas se emplean mediante el navegador web (Cobo, Gómez, Pérez, & Rocha, 2005).

#### **Lenguajes del lado del cliente**

Los lenguajes de programación del lado del cliente se usan para su integración en páginas web. Un código escrito en un lenguaje de script se incorpora directamente dentro de un código HTML (*HyperText Markup Language*) y se ejecuta interpretado, no compilado. Con la programación del lado del cliente se

pueden validar algunos de los datos en la máquina cliente antes de enviarlos al servidor. Esto proporciona a los usuarios informes de error inmediatos, mientras siguen en esa página de formulario y sin necesidad de volver atrás luego de recibir un mensaje de error (Torre, 2015).

## **HTML 5**

HTML es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con enlaces que conducen a otros documentos o fuentes de información relacionadas y con inserciones multimedia (gráficos, sonido).

El mismo proporciona una plataforma con la cual desarrollar aplicaciones web parecidas a las aplicaciones de escritorio, donde su ejecución dentro de un navegador no implique falta de recursos o facilidades para resolver las necesidades reales de los desarrolladores. HTML 5 es la versión a utilizar para la implementación del sistema por las mejoras en la creación de la estructura del código web y en el manejo óptimo de las etiquetas. Algunas de estas mejoras son (LaGrone, 2013):

- Nueva estructura de etiquetas mejorada: esta nueva estructura permite definir por separado el encabezado, la barra de navegación, las secciones y el pie de página.
- Ofrece la posibilidad de obtener un código más limpio y fácil de depurar que los códigos de los estándares anteriores.
- Cuenta con nuevos tipos de campos para realizar validaciones, tales como: el atributo “pattern” en el uso de expresiones regulares para personalizar reglas de validación y el atributo “required” indicando que el campo es obligatorio.
- Mayor variedad en tipos de campos de los formularios con respecto a versiones anteriores.

## **CSS 3**

Hojas de Estilo en Cascada (Cascading Style Sheets o CSS) es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etc. La filosofía de CSS se

basa en separar los contenidos de la presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). De esta forma mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (Richard, Studholme, Murphy, & Manian, 2012).

La razón por la cual se utilizará CSS3 es porque ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, lo que hace necesaria su utilización.

### **JavaScript**

JavaScript es un lenguaje de programación que surgió con el objetivo inicial de programar ciertos comportamientos sobre las páginas web, respondiendo a la interacción del usuario y la realización de automatismos sencillos. En ese contexto se puede decir que nació como un "lenguaje de scripting" del lado del cliente, sin embargo, hoy es mucho más. Las necesidades de las aplicaciones webs modernas y el HTML 5 han provocado que el uso de JavaScript haya llegado a unos niveles de complejidad y prestaciones tan grandes como otros lenguajes de primer nivel (Tilkov & Vinoski, 2010).

Se utiliza del lado del cliente (se ejecuta en el ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los documentos HTML permiten incrustar fragmentos de código JavaScript, bien dentro del propio archivo HTML o realizando una carga externa del archivo indicando donde se encuentra el código JavaScript. Dentro de un documento HTML puede haber ninguno, uno o varios scripts de JavaScript. Este lenguaje posee varias características, entre ellas se pueden mencionar que es un lenguaje basado en acciones que posee menos restricciones. El uso de este lenguaje es muy habitual en la programación web (Tilkov & Vinoski, 2010).

La selección de HTML 5, CSS 3 y JavaScript como lenguajes del lado del cliente para esta investigación es por la gran variedad de opciones y ventajas

que ofrecen en la actualidad en los diseños web. Además de constituir la base de los frameworks que se utilizarán para el desarrollo de la aplicación.

#### 1.8.4 Framework del lado del cliente.

**Bootstrap 4.0** es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y que se visualicen de forma correcta. Es Open Source o código abierto, por lo que se puede usar de forma gratuita y sin restricciones (Mestras, 2011).

La más genérica es que permite simplificar el proceso de maquetación, sirviendo de guía para aplicar las buenas prácticas y los diferentes estándares. Estas son algunas ventajas de su utilización (Mestras, 2011):

- Se puede tener una web rápidamente bien organizada de forma visual si existen conocimientos de maquetación.
- Permite utilizar muchos elementos web: desde iconos a desplegables, combinando HTML 5, CSS 3 y JavaScript.
- El diseño será adaptable, no importa el dispositivo, la escala o resolución.
- El *grid system*: maquetar por columnas, es sencillo y muy configurable.
- Se integra muy bien con las principales librerías JavaScript.
- Curva de aprendizaje baja.
- Contiene varios componentes del diseño como: glyphicons (íconos), ventanas de diálogo que requieren una respuesta sencilla del usuario (modal), mensajes de alerta, dropdowns (menú desplegable), botones, breadcrumb (indica el lugar de navegación en la jerarquía del sitio), paginado, entre otros.
- El haber sido creado por Twitter brinda ciertas garantías: está muy pensado y hay mucho trabajo ya hecho. Por lo tanto, hay una comunidad muy activa creando, arreglando, ofreciendo plugins y mucho más.

- Tipografía en “ems” o “rems”, lo cual facilitará un comportamiento responsive y ayudará cuando se aplique *media queries* a los textos.

## **jQuery 1.12**

**jQuery:** es un framework para el lenguaje JavaScript, fue creada por John Resig, que permite simplificar la programación en este lenguaje. Implementa una serie de clases (de Programación Orientada a Objetos (POO)) que permiten programar sin preocuparse del navegador con el que el usuario está visualizando la página, ya que funcionan igual en todas las plataformas más habituales. Ofrece una infraestructura con la que se tiene mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con jQuery se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Le ofrece a los desarrolladores todas sus ventajas de forma gratuita, ya que el framework tiene licencia para uso en cualquier tipo de plataforma, personal o comercial (Community, 2010).

### **1.8.5 Entorno de desarrollo integrado.**

Un entorno de desarrollo integrado o IDE (acrónimo en inglés de *integrated development environment*), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI) (Corporation, 2015).

## **Netbeans 8.0**

**Netbeans** es un entorno de desarrollo gratuito y de código abierto. Permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones web, o para dispositivos móviles. Da soporte a las siguientes tecnologías: Java, PHP, Groovy, C/C++ y HTML 5. Además, puede instalarse en varios sistemas operativos: Windows, Linux y Mac OS. Buen editor de código, multilenguaje, con el habitual coloreado, sugerencias de código, comprobaciones sintácticas y semánticas. Permite localizar la ubicación de la clase actual. Posee plantillas de código que agilizan el proceso de desarrollo y

herramientas de refactorización que es una técnica de la ingeniería de software para reestructurar un código fuente, alterando su estructura interna sin cambiar su comportamiento externo (Rodríguez, Siles, Quijano, Gutiérrez, & Álvarez, 2014).

## CAPÍTULO 2: DESCRIPCIÓN Y CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### 2.1 Introducción del capítulo.

En el presente capítulo se realiza una descripción de las características principales del sistema a desarrollar de acuerdo al problema por el cual fue concebido. Se especifican las funcionalidades que se desean informatizar en la búsqueda de satisfacer las necesidades del cliente, además se describe la construcción de la propuesta con el desarrollo de las historias de usuario a través de iteraciones. Se definen tareas y principios de diseño para su implementación. Se exponen pruebas de aceptación para garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable.

### 2.2 Personas relacionadas con el sistema.

Las personas relacionadas con el sistema, son aquella que de una forma u otra interactúan con éste y obtienen un resultado de uno o varios procesos que en él se ejecutan, además de aquellas que se encuentran involucradas en dichos procesos, participan en ellos, pero no obtienen ningún resultado de valor. A continuación, en la **tabla 2.1** se muestran las personas relacionadas con el sistema.

Tabla 2.1 Personas relacionadas con el sistema.

Persona	Justificación
Usuario: (profesor o estudiante)	Representa la persona que va a interactuar con el sistema para realizar las operaciones con matrices, visualizar operaciones, ver los materiales y acceder a la ayuda del sistema.

### 2.3 Exploración

La metodología de desarrollo XP comienza con la fase de exploración. En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de

desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología (Joskowicz, 2008b).

### 2.3.1 Historias de usuario.

Las “Historias de usuarios” (HU) sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. 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 (Joskowicz, 2008b). A continuación, se muestran las HU

Tabla 2.2 Historias de usuario "Sumar matrices"

Historia de usuario	
<b>Número:</b> 1	<b>Usuario:</b> (profesor o estudiante)
<b>Nombre historia:</b> Sumar matrices.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.3	<b>Iteración asignada:</b> 1
<b>Programadores responsables:</b>	
<b>Descripción:</b> Inicia cuando el usuario accede a la aplicación, se le brinda la posibilidad de que introduzca sus datos ( <i>usuario y contraseña</i> ), con el objetivo de comprobar si está registrado y así poder asignarle los permisos según su rol en el sistema.	
<b>Observaciones:</b> Es necesario estar registrado en el sistema para poder	

acceder a las funcionalidades que le correspondan según el nivel que tenga asignado.

Tabla 2.3 Historias de usuario " Visualizar suma de matrices"

Historia de usuario	
<b>Número:</b> 9	<b>Usuario:</b> (profesor o estudiante)
<b>Nombre historia:</b> Visualizar suma de matrices	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Media
<b>Puntos estimados:</b> 0.5	<b>Iteración asignada:</b> 2
<b>Programadores responsables:</b>	
<b>Descripción:</b> Inicia cuando el usuario accede a la aplicación, escoge la opción Visualización para visualizar la suma de matrices. Selecciona el tamaño de las matrices y da clic en el botón crear matrices. Llena el formulario para sumar las matrices, luego da clic en el botón calcular y se realiza la operación. El usuario selecciona el tiempo de visualización si lo desea y termina cuando el usuario da clic en el botón paso a paso para ver la visualización de la operación.	
<b>Observaciones:</b> Es necesario tener en cuenta las condiciones para realizar la operación y llenar correctamente los campos.	

#### 2.4 Restricciones que el sistema debe cumplir.

Las condiciones que el sistema debe cumplir o capacidad que debe tener con el objetivo de establecer un entendimiento común entre el usuario y el proyecto de software son los requerimientos. El propósito de su gestión es establecer un entendimiento común entre el usuario y el desarrollador de software.

#### Requerimientos no funcionales

Los requerimientos no funcionales son requerimientos de calidad y son restricciones. Sobre esta base se realiza un estudio comparativo de sus enfoques de desarrollo de aplicaciones web. Se estudian los procesos de desarrollo que cada una de ellas utiliza para determinar los requerimientos no funcionales; las técnicas de la ingeniería de requerimientos que proponen para su especificación, validación y gestión; y se establece en qué fase del ciclo de vida de desarrollo de software identifican y tratan a los requerimientos no funcionales (P. D. R. S. Pressman, 2010).

### **Apariencia o interfaz externa**

- Diseño sencillo, orientado al entorno de trabajo del cliente para que se sienta identificado con la aplicación.
- Los colores deben ser claros y agradables a la vista del usuario.
- El sistema debe tener una interfaz cómoda, facilitando la navegación y evitando la utilización en exceso de imágenes.
- El vocabulario utilizado será en idioma español exclusivamente.

### **Usabilidad**

- El sistema será utilizado por cualquier estudiante o profesor desde cualquier sede de la Universidad ya que no requiere autenticación.

### **Software**

- Las computadoras clientes deben contar con Internet Mozilla Firefox 3.2 o superior.
- El sistema informático es multiplataforma por lo que corre en cualquier sistema operativo.

### **Hardware**

- Las computadoras clientes deben tener 128 MB de RAM o superior.
- Las computadoras clientes deben estar conectadas a la red.

### **Seguridad**

### **Confiabilidad**

- La información almacenada en el sistema debe corresponderse a las diferentes operaciones con matrices que realiza el software.

## 2.5 Planificación.

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Típicamente esta consiste en una o varias reuniones grupales de planificación. Para la planificación del proyecto se en cuenta una de las buenas prácticas de la metodología XP, la cual plantea que se deben trabajar 40 horas semanales. Esto significa que la semana tendrá 5 días. El resultado de esta es un plan de entregas (Joskowicz, 2008b).

### 2.5.1 Estimación de esfuerzos por historias de usuarios.

Para el desarrollo de la aplicación propuesta en este trabajo se realizó una estimación del esfuerzo para cada una de las historias de usuario identificadas, permitiendo tener una medida real de la velocidad de progreso del proyecto y brindando una guía razonable a la cual ajustarse, llegándose así a los resultados que se muestran en la siguiente tabla (Joskowicz, 2008).

Tabla 2.4 Estimación de esfuerzos por historias de usuarios.

No	Historia de usuario	Prioridad	Riesgo	Esfuerzo	Iteración
1	Sumar matrices.	Alta	Media	0.3	1
2	Restar matrices.	Alta	Media	0.3	1
3	Multiplicar un escalar por una matriz.	Alta	Media	0.5	1
4	Multiplicar dos matrices.	Alta	Media	0.5	1
5	Hallar la traspuesta de una matriz.	Alta	Media	0.3	1

<b>6</b>	Calcular determinante.	Alta	Media	0.5	1
<b>7</b>	Escalonar matriz por el método de pivote parcial.	Alta	Media	0.3	1
<b>8</b>	Escalonar matriz por el método de pivote elemental.	Alta	Media	0.3	1
<b>9</b>	Visualizar suma de matrices.	Alta	Media	0.5	2
<b>10</b>	Visualizar resta de matrices	Alta	Media	0.4	2
<b>11</b>	Visualizar multiplicación de un escalar por una matriz.	Alta	Media	1.4	2
<b>12</b>	Visualizar multiplicación de dos matrices.	Alta	Media	1.4	2
<b>13</b>	Visualizar traspuesta de una matriz.	Alta	Media	1.3	2
<b>14</b>	Visualizar matriz escalón por el método de pivote parcial.	Alta	Media	1.4	3
<b>15</b>	Visualizar matriz escalón por el método de pivote elemental.	Alta	Media	1.4	3
<b>16</b>	Hallar la inversa de una matriz.	Alta	Media	1.4	3

<b>17</b>	Mostrar materiales	Alta	Media	0.4	3
<b>18</b>	Mostrar Ayuda.	Alta	Media	0.4	3

### 2.5.2 Plan de duración de las iteraciones.

En esta fase, se identifican varias iteraciones sobre el sistema antes de ser entregado, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración (Joskowicz, 2008).

De acuerdo a lo mencionado anteriormente, se decidió realizar dicha planificación en tres iteraciones, las cuales se detallan a continuación.

Tabla 2.5 Plan de duración de las iteraciones.

<b>Iteración</b>	<b>Historias de usuario</b>	<b>Duración de la iteración</b>
Iteración 1	<ol style="list-style-type: none"> <li>1. Sumar matrices.</li> <li>2. Restar matrices.</li> <li>3. Multiplicar un escalar por una matriz.</li> <li>4. Multiplicar dos matrices.</li> <li>5. Hallar la traspuesta de una matriz.</li> <li>6. Calcular determinante.</li> <li>7. Escalonar matriz por el método de pivote parcial.</li> <li>8. Escalonar matriz por el método de pivote elemental.</li> </ol>	6 semanas

Iteración 2	<p>9. Visualizar suma de matrices.</p> <p>10. Visualizar resta de matrices.</p> <p>11. Visualizar multiplicación de un escalar por una matriz.</p> <p>12. Visualizar multiplicación de dos matrices.</p> <p>13. Visualizar traspuesta de una matriz.</p>	7 semanas
Iteración 3	<p>14. Visualizar matriz escalón por el método de pivote parcial.</p> <p>15. Visualizar matriz escalón por el método de pivote elemental.</p> <p>16. Hallar la inversa de una matriz.</p> <p>17. Mostrar materiales.</p> <p>18. Mostrar Ayuda.</p>	7 semanas

### 2.5.3 Plan de entrega.

El cronograma de entregas establece que las 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 (cliente, desarrolladores y gerentes). XP denomina a esta reunión “Juego de planeamiento” (“Planning game”), pero puede denominarse de la manera que sea más apropiada al tipo de empresa y cliente. En la siguiente tabla se muestra el plan de duración de entregas en el cual se especifica un aproximado de las fechas para cada iteración (Joskowicz, 2008b).

Tabla 2.6 Plan de entregas.

Iteración	Iteración 1	Iteración 2	Iteración 3
Entrega	Final 1ra Iteración 2da semana de febrero	Final 2da Iteración 5ta semana de marzo	Final 3ra iteración 3ra semana de mayo

## 2.6 Iteraciones.

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas 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 (Joskowicz, 2008b).

### 2.6.1 Tareas.

En esta fase XP plantea la implementación de cada una de las historias de usuarios. Al principio, se lleva a cabo una revisión del 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. En cada iteración XP propone la realización de varias tareas asignadas cada una a un programador específico (Joskowicz, 2008b).

A continuación se presentan algunas de las tareas perteneciente a las diferentes iteraciones definidas, el resto se pueden consultar en el Expediente del proyecto.

Tabla 2.7 Tarea Mostrar formulario para sumar matrices.

Tarea	
<b>Número de tarea:</b> 1	<b>Número de historia:</b> 1
<b>Nombre de tarea:</b> Mostrar formulario para sumar matrices.	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0.3
<b>Fecha inicio:</b> 1/01/2020	<b>Fecha fin:</b> 03/01/2020
<b>Programador responsable:</b>	

<b>Descripción:</b> diseñar una interfaz que permita al usuario insertar los datos necesarios para sumar matrices.
--

Tabla 2.8 Tarea Visualizar suma de matrices

Tarea	
<b>Número de tarea:</b> 1	<b>Número de historia:</b> 9
<b>Nombre de tarea:</b> Visualizar suma de matrices	
<b>Tipo de tarea:</b> Desarrollo.	<b>Puntos estimados:</b> 0.5
<b>Fecha inicio:</b> 12/02/2020	<b>Fecha fin:</b> 16/02/2020
<b>Programador responsable:</b>	
<b>Descripción:</b> diseñar una interfaz que permita al usuario visualizar paso a paso la suma de matrices.	

## 2.7 Producción.

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento) (Joskowicz, 2008b).

### 2.7.1 Diseño.

Este sistema fue diseñado, de forma dinámica y agradable al usuario. Para lograrlo se emplearon algunos principios de diseño visual en las páginas web que la conforman. La aplicación presenta un diseño simple y sencillo, orientado

al entorno de trabajo del cliente para que se sienta identificado con la aplicación. Los colores que se utilizaron fueron el blanco y el gris principalmente (**Joskowicz, 2008b**).

Para la construcción del sistema se tomaron en cuenta algunas reglas para el diseño: un header o banner, donde se muestra la información general de sistema como logo de la institución, el nombre del sistema e imágenes y textos que muestren de manera general el contenido de la aplicación, un menú en la parte izquierda donde se encuentran los diferentes vínculos de gestión del sistema y en la parte inferior izquierda se muestran los accesos a las secciones del sistema, la sección del contenido donde se muestra la información que se desea buscar.

Dentro del estilo arquitectónico llamada y retorno se encuentra el patrón **Modelo-Vista-Controlador**: es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el **modelo**, la **vista** y el **controlador**, es decir, por un lado, define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. Para un mejor entendimiento de esta arquitectura se explican a continuación los tres componentes:

- El **Modelo**: es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación (lógica de negocio). Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada (típicamente a un usuario). Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.
- El **Controlador**: responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la

información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta de 'modelo' (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos), por tanto, se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.

- La **Vista**: presenta el 'modelo' (información y *lógica de negocio*) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida (Jurado, 2008).



Figura 3: Representación del patrón de arquitectura Modelo-Vista-Controlador

### Patrones de diseño

Un patrón de diseño es una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (P. D. R. S. Pressman, 2010).

Para el sistema que se desea implementar se seleccionan los patrones generales de software para la asignación de responsabilidades (en inglés General Responsibility Assignment Software Patterns (GRASP)) los cuales se encuentran dentro de los patrones de diseño de comportamiento, o sea, están dirigidos a cómo debe actuar el sistema ante el usuario final.

Los patrones GRASP tienen varias clasificaciones, para el desarrollo del sistema se utilizarán tres de estas clasificaciones, las cuales son:

**Bajo acoplamiento:** el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con qué las conoce y con qué recurre a ellas. En tal sentido, el término bajo acoplamiento significa que una clase no depende de muchas clases. Lo anterior permite proteger a una clase de los cambios que ocurran en las otras con las cuales se relaciona (Fowler, 2003).

**Alta cohesión:** es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase (P. D. R. S. Pressman, 2010). Plantea que la información que almacena cada clase debe ser coherente y estar relacionada con la misma.

Aparejado al bajo acoplamiento se encuentra la alta cohesión, teniendo en cuenta que las clases del sistema están diseñadas para que desempeñen una labor única, pudiéndose dedicar a un propósito específico. Esto facilita que las clases sean fáciles de entender y de reutilizar.

**Experto en información:** es el encargado de asignar la responsabilidad de la creación de un objeto o la implementación de un método a una clase que contenga toda la información necesaria para cumplir con esta responsabilidad (P. D. R. S. Pressman, 2010). De esta manera se garantiza la obtención de un diseño con mayor cohesión en su estructura, robusto, fácil de mantener y con la información centralizada en una misma clase.

El diseño de la base de datos fue realizado con la herramienta DBDesigner4, la misma está compuesta por 17 tablas, las cuales están

normalizadas, cumpliendo con las normas establecidas para el diseño de bases de datos. Ver el Anexo III con el diseño de la base de datos.

### **2.7.2. Codificación.**

El lenguaje de programación que se utilizó fue PHP, el cual es un lenguaje del lado del servidor y fue diseñado originalmente para la creación de aplicaciones web dinámicas, se utilizó código JQuery que a su vez proporciona una plataforma sólida en la cual construir una interfaz rica e interconectada con máxima compatibilidad y estabilidad, JavaScript para las vistas con el gestor de base de datos MySQL por la magnitud y exigencia para desarrollar la aplicación. HTML 5, es el lenguaje de marcado predominante para la creación y publicación de páginas web, al que se le pueden añadir características y funcionalidades mediante CSS, este último se encarga de incorporar mejoras centradas en el ámbito visual en otras palabras HTML es el que se va a mostrar, y CSS es como se va a mostrar. Además, se tuvo en cuenta la reutilización de código pues es una manera de agilizar la producción del desarrollo del software

#### **Estándares de codificación**

Un estándar de codificación comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez. Cuando el proyecto de software incorpore código fuente previo, o bien cuando realice el mantenimiento de un sistema de software creado anteriormente, el estándar de codificación debería establecer cómo operar con la base de código existente (Joskowicz, 2008b).

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido y se utilizan técnicas de programación apropiadas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener.

#### **Declaraciones de variables**

- Las variables deben ser declaradas justo antes de ser utilizadas.
- Cuando sea posible, la inicialización de las variables locales ocurrirá en el momento de su declaración.
- Las variables comienzan con el signo de \$ seguido del nombre.

### **Espacio en blanco**

El uso de espacios en blanco mejora la legibilidad del código agrupando secciones de código que se relacionan lógicamente. Se recomienda usar espacios en blanco entre los operadores lógicos y aritméticos para lograr una mayor legibilidad en el código.

### **Sentencias SQL**

Todas las letras en mayúscula. Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

### **Declaraciones de clases**

Para la declaración de clases se siguieron las siguientes reglas:

- Ningún espacio en blanco entre el nombre de un método y el paréntesis “(” que abre su lista de parámetros.
- La llave de apertura “{” aparece en una nueva línea sin más instrucciones.
- La llave de cierre “}” empieza en una nueva línea endentada para ajustarse a su sentencia de apertura correspondiente, excepto cuando no existen sentencias entre ambas, que debe aparecer inmediatamente después de la de apertura “{”.
- Los métodos se separan con una línea en blanco.

## CONCLUSIONES GENERALES

1. En la investigación ha quedado expuesto todo lo referente a las bases teóricas del proceso de enseñanza-aprendizaje de la asignatura de Álgebra Lineal en la carrera de Ingeniería Informática. El estudio de los sistemas similares existentes, arrojó como resultado que todos se caracterizan por la simplicidad en la representación de la información, su fácil manejo y un diseño intuitivo, todas estas características pueden ser tomadas en cuenta para ser incorporadas en el nuevo sistema informático que se pretende desarrollar.
2. El diseño del sistema informático para el trabajo con matrices en la asignatura Álgebra Lineal, permite la creación de una aplicación flexible y reutilizable. El estándar de codificación definido permite un código legible que garantiza un mejor mantenimiento del sistema y el desarrollo de futuras versiones. Se ha documentado la aplicación con la utilización de la metodología Programación Extrema XP, con la finalidad de lograr una adecuada planificación, análisis y diseño e implementación de la aplicación y así; generar valor con cada entregable al final de cada iteración.

## **RECOMENDACIONES**

Luego de haber analizado los resultados de la presente investigación se recomienda:

Desarrollar el sistema informático para el trabajo con matrices en la asignatura Álgebra Lineal de la carrera Ingeniería Informática en la Universidad de Guantánamo.

## REFERENCIAS BIBLIOGRÁFICAS

1. @Manz. (2018). Lenguaje Javascript - Documentación sobre programación web. Obtenido de <https://lenguajejs.com/>
2. Aliaga, Z. L. (2012). Sistemas de objeto de aprendizaje para favorecer la apropiación de la implementación de clases en la asignatura Ingeniería de Software II (Tesis de Maestría). Universidad de las ciencias informáticas, Manzanillo, Guantánamo.
3. Acedo, J. I. (4 de mayo de 2015). *Apuntes de Programación*. Obtenido de <http://programacion.jias.es/2015/05/web-¿que-es-el-framework-bootstrap>.
4. Almaguel, A. (2008). Software educativo para apoyar el aprendizaje de operaciones con matrices en los estudiantes de ingeniería. Universidad de Guantánamo, Bayamo, Guantánamo.
5. Álvarez, C. (2010). Como se modela la investigación científica. (1ra Edición., Vol. 1ra Parte.). Habana, Cuba.
6. Apache Friends. (2019). XAMPP Installers and Downloads for Apache Friends. Obtenido de <https://www.apachefriends.org/es/index.html>
7. Augustine, S. (2005). *Managing Agile Projects*. Prentice Hall PTR.
8. Barbolla, P. S. (1999). *Álgebra lineal y teoría de matrices*.
9. Barzaba, C. (2006). Metodologías de desarrollo de software. Retrieved from <http://www.um.es/docencia/barzana/IAGP/lagp2.html#BM1>
10. Bootstrap team. (2019). Bootstrap · The most popular HTML, CSS, and JS library in the world. Obtenido de <https://getbootstrap.com/>
11. Bonilla, A., Mendoza, D., & Lavidalie, R. (2005). Propuesta hacia la iniciativa para la sociedad de la Información en Guatemala. Retrieved from [http://www.colombiadigital.net/newcd/component/docman/doc\\_download/1852](http://www.colombiadigital.net/newcd/component/docman/doc_download/1852).
12. Cayley, A. (1858). *Matrices*. Retrieved from [http://personal5.iddeo.es/ztt/Tem/T6\\_Matrices.htm](http://personal5.iddeo.es/ztt/Tem/T6_Matrices.htm).

13. Cervantes, D. H. (2018). *SG Buzz*. Obtenido de Arquitectura de Software: <https://sg.com.mx/revista/27/arquitectura-software>
14. Cobo, A., Gómez, P., Pérez, D., & Rocha, R. (2005). PHP y MySQL. Tecnologías para el desarrollo de aplicaciones web.
15. Community, E. (2010). jQuery CookBook.
16. Corporation, L. (2015). Entornos de Desarrollos Integrados. Retrieved from <http://es.slideshare.net/GhaBiithahh/entornos-de-desarrollo-integrados>.
17. Conallen, J. (2003). Building Web Application with UML
18. EcuRed. (2019). *EcuRed: Enciclopedia Cubana*. Recuperado el 07 de Mayo de 2019, de [https://www.ecured.cu/Herramienta\\_CASE](https://www.ecured.cu/Herramienta_CASE)
19. Erich Gamma, R. H., Ralph Johnson, John Vlissides (2013). "Design Patterns: Elements of Reusable."
20. Fowler, M. (2003). Patterns of Enterprise Application Architecture.
21. García, E. C. R. (2004). Recursos computacionales para la enseñanza aprendizaje de la matemática en la educación superior.
22. García, & Gertrudix, M. (2011). Naturaleza y características de los servicios y los contenidos digitales abiertos. CIC Cuadernos de Información y Comunicación.
23. Guido, S. (1997). Álgebra Lineal.
24. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1997). Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley.
25. ISSI, G. (2003). Metodologías Ágiles en el Desarrollo de Software.
26. Joskowicz. (2008a). Reglas y Prácticas en eXtreme Programming.
27. Joskowicz, I. J. (2008b). Reglas y Prácticas en eXtreme Programming. Retrieved from <http://iie.fing.edu.uy/~josej/docs/XP%20-20Jose%20Joskowicz>.
28. Jurado, J. L. (2008). PATRON DE DISEÑO MVC (Modelo Vista Controlador). (F. de I. E. y Telecomunicaciones, Trans.) (p. 3).
29. Jacobson, I., Booch, G., & Rumbaugh, J. (2000). UML y Patrones. Prentice Hall.
30. Kruchten, P. (2011). The Rational Unified Process--An Introduction. University of British Columbia - Vancouver: Rational Software.

31. Kadlec, T. (2013). *Implementing Responsive Design: Building sites for an anywhere, everywhere web*. Eighth Street Berkeley, United States of America: New Riders. Obtenido de [www.newriders.com](http://www.newriders.com)
32. Larman, C. (1999). *Applying UML and Patterns An Introduction to Object-Oriented Analysis and Design* (Primera ed.). EUA: Prentice Hall, Inc.
33. LaGrone, B. (2013). *HTML5 and CSS3 Responsive Web Design Cookbook*. San Francisco, California, USA: PACKT.
34. Lamas, R. R., & García, D. (2000). *Introducción a la informática educativa*. ISPJAE, Cuba.
35. La Revista Informática.com. (2016). *Lenguaje de Programación JavaScript*. *La Revista Informática.com*. Obtenido de <http://www.larevistainformatica.com/JavaScript.htm>
36. León, R. A. H., & González, S. C. (2002). *El paradigma cuantitativo de la investigación científica*. Ciudad de La Habana: Eduniv, Editorial universitaria.
37. Lozano, R. D. (2009). *Enfoque histórico de la enseñanza de las matemáticas*. Docente de la Universidad Libre de Cali, Colombia.
38. Magazine, F. (2015). *Ventajas de las aplicaciones web*. Retrieved from <http://www.pixima.net/aplicaciones-web/ventajas-de-las-aplicaciones-web/>.
39. Marqués, P. (1999). *Software educativo multimedia: tipologías*. Retrieved from [http://www.lmi.ub.es/te/any96/marques\\_software/](http://www.lmi.ub.es/te/any96/marques_software/)
40. Matthew B. Hoy, "HTML5: A New Standard for the Web," *Medical Reference Services Quarterly*, vol. 30:1, 2015.
41. Mestras, J. P. (2011). *Aplicaciones Web/Sistemas Web*. Bootstrap 3.0.
42. NeoAttack. (2018). *NeoAttack*. Obtenido de *¿Qué es un Framework y para que sirve? - Neo Wiki*: <https://neoattack.com/neowiki/framework/>
43. Ortega, A. R. (2003). *Ambientes innovadores de aprendizaje*.
44. Pressman, P. D. R. S. (2010). *SOFTWARE ENGINEERING. A PRACTITIONER'S APPROACH*.
45. P. Lubbers, B. Albers, F. Salim, P. Lubbers, B. Albers, and F. Salim, "Using the html5 web storage api," in *Pro HTML5 Programming*, pp. 213–241, Apress, 2010.10.1007/978-1-4302-2791-5 9.

46. Pressman, R. S. (2010). *Software engineering. A practitioner's Approach.* (7ma ed.). McGraw-Hill.
47. Richard, C., Studholme, O., Murphy, C., & Manian, D. (2012). *Beginning HTML5 and CSS3.* New York.
48. Rodríguez, R., Codina, L., & Pedraza, R. (2010). *Cibermedios y web 2.0: modelo de análisis y resultados de aplicación.* El profesional de la información.
49. Rodríguez, Siles, F., Quijano, J., Gutiérrez, P., & Álvarez, C. (2014). *NetBeans.* Retrieved from <http://www.genbetadev.com/herramientas/netbeans-1>.
50. Salvador, G. U. (1998). *Breves de Álgebra Lineal.*
51. Sánchez, J. (1999). *Construyendo y aprendiendo el computador.* Santiago de Chile, Universidad de Chile.
52. Schwaber, K. (2010). *Advanced Development Methods. SCRUM Development Process.*
53. Streetdirectory. (2015). *Desktop Applications Vs. Web Applications.* Retrieved from [http://www.streetdirectory.com/travel\\_guide/114448/programming/desktop\\_applications\\_vs\\_web\\_applications.html](http://www.streetdirectory.com/travel_guide/114448/programming/desktop_applications_vs_web_applications.html)
54. Tilkov, S., & Vinoski, S. (2010). *Node. js: Using JavaScript to build high-performance network programs.* IEEE Internet Computing.
55. Tecnologías, U. I. (20 de 01 de 2016). *Bootstrap, Ventajas y Desventajas.* Recuperado el 12 de 12 de 2018, de *Bootstrap, Ventajas y Desventajas:* <http://tecnologiaenvivo.com/bootstrap-ventajas-y-desventajas/>
56. The jQuery Foundation. (2019). *jQuery.* Obtenido de <https://jquery.com/>
57. Torre, A. D. L. (2015). *Lenguajes de programación.* Retrieved from <http://www.es.ccm.net/s/lenguaje+de+programacion?qlc>
58. UNIWEBSIDAD. (08 de 12 de 2017). *Introducción a CSS.* Recuperado el 25 de 02 de 2018, de *Introducción a CSS:* <https://uniwebsidad.com/libros/css/capitulo-1>.
59. Urbaneja, P. E. (2002). *La historia de la Matemática como recurso didáctico e instrumento de integración cultural de la Matemática.*

60. Zakas, N. C. (2005). Professional JavaScript™ for Web Developers. (1ra Edición). Indianapolis, Indiana.

## **GLOSARIO DE TÉRMINOS**

CSS: Cascading Style Sheets

HTML: HyperText Markup Language

PC: Computadora Personal.

RAM: *Random-Access Memory* o memoria de acceso aleatorio.

TIC: Tecnologías de Información y las Comunicaciones

XP: Extreme Programming