



**UNIVERSIDAD NACIONAL
“PEDRO RUIZ GALLO”**



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

ESCUELA PROFESIONAL DE COMPUTACIÓN E INFORMÁTICA

**SISTEMA DE ANÁLISIS ESTRUCTURAL EN PLATAFORMA
JAVA PARA VIGAS CONTINUAS POR EL MÉTODO DE
ELEMENTOS FINITOS**

TESIS:

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO EN COMPUTACIÓN
E INFORMÁTICA**

AUTORES:

AGUILAR NEVADO, JOSÉ LUIS

EXEBIO TANTARICO, JOSÉ ANTONIO

LAMBAYEQUE, ENERO DEL 2017

**SISTEMA DE ANÁLISIS ESTRUCTURAL EN PLATAFORMA JAVA PARA VIGAS
CONTINUAS POR EL MÉTODO DE ELEMENTOS FINITOS**

**TESIS PARA OPTAR EL TÍTULO DE INGENIERO EN COMPUTACIÓN E
INFORMÁTICA, QUE PRESENTAN
LOS BACHILLERES:**

Bach. JOSÉ LUIS AGUILAR NEVADO

Responsable

Bach. JOSÉ ANTONIO EXEBIO TANTARICO

Responsable

ASESOR:

Ing. ALEJANDRO CHAYAN COLOMA

Asesor

**SISTEMA DE ANÁLISIS ESTRUCTURAL EN PLATAFORMA JAVA PARA VIGAS
CONTINUAS POR EL MÉTODO DE ELEMENTOS FINITOS**

**TESIS PARA OPTAR EL TÍTULO DE INGENIERO EN COMPUTACIÓN E
INFORMÁTICA**

APROBADA POR EL JURADO:

Ing. Luis Alberto Reyes Lescano
Presidente

Mg. Ing. Gilberto Carrión Barco
Secretario

Ing. Percy Celis Bravo
Vocal

RESUMEN

En la actualidad el ser humano está en contacto constante con la tecnología, ya sea en el hogar, en el trabajo o en la escuela. Las redes sociales, los teléfonos inteligentes, las tabletas y los libros electrónicos son ejemplos de productos tecnológicos que vemos todos los días. En los últimos años, se ha dado un gran avance en cuanto a la facilidad de uso de las tecnologías de la información en el campo de la educación. Con la intención de ser utilizados en el contexto del proceso de enseñanza–aprendizaje, la Universidad Cesar Vallejo con sede Chiclayo no escapa de la posibilidad de ofrecer a sus estudiantes dichas tecnologías.

Al aplicar la encuesta tanto a los estudiantes de la carrera de ingeniería civil como al personal docente que labora en dicha institución se determinó como problemática en estudio, la escasez de material educativo para el análisis estructural en plataforma java para el análisis estructural por el método de elementos finitos.

Para darle solución al problema, se planteó el desarrollo de un Material Educativo Computarizado usando la metodología de Álvaro Galvis y de complemento se realizó el análisis y diseño aplicando UML.

El presente trabajo de tesis se estructura en cuatro partes:

Una primera parte sobre la introducción, presentando: el planteamiento del problema, y estableciéndose el objetivo general y específicos para dar solución al problema, importancia, el alcance del proyecto y el marco referencial.

Las técnicas aplicadas en la investigación, la muestra de la población y el desarrollo de la propuesta se encuentran en la segunda parte de material y métodos.

Mientras que en la tercera parte se encuentra los resultados y discusión, en la cuarta se encuentra las conclusiones y recomendaciones.

ABSTRACT

At present, the human being is in constant contact with technology, whether at home, at work or at school. Social networks, smartphones, tablets and e-books are examples of technological products we see every day. In recent years, there has been a breakthrough in the ease of use of information technologies in the field of education. With the intention of being used in the context of the teaching-learning process, the Cesar Vallejo University based in Chiclayo does not escape the possibility of offering its students such technologies.

When applying the survey to both students of the civil engineering career and to the teaching staff working in that institution, the study of the shortage of educational material for the structural analysis in the Java platform for structural analysis by the method of finite elements.

In order to solve the problem, it was proposed the development of a Computerized Educational Material using the methodology of Álvaro Galvis and complemented the analysis and design using UML.

The present work of thesis is structured in four parts:

A first part on the introduction, presenting: the problem approach, and establishing the general and specific objective to solve the problem, importance, scope of the project and the frame of reference.

The techniques applied in the research, the sample of the population and the development of the proposal are in the second part of material and methods.

While in the third part you will find the results and discussion, in the fourth you will find the conclusions and recommendations.

DEDICATORIAS

En primer lugar, agradecer a Dios por darme fuerza, salud e inteligencia para poder llegar hasta este momento tan importante en mi vida.

En segundo lugar, agradecer a mi padre José de la Cruz Aguilar Díaz y a mi madre Teresa Nevado Panta; por su constante apoyo y preocupación por que alcancé mis metas, sueños y objetivos desde que empecé el colegio, pasando por la universidad y por todo lo que me tiene preparado el futuro.

Finalmente agradecer a mis amigos y profesores, pues gracias a ellos la vida universitaria ha sido tan emocionante y amena.

De José Luis Aguilar Nevado

A:

Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante toda mi vida.

Mi madre Manuela, por darme la vida, quererme mucho, creer en mí y porque siempre me apoyaste.
Mamá gracias por darme una carrera, un futuro, todo esto te lo debo a ti.

De Exebio Tantarico, Jose Antonio

INTRODUCCIÓN	1
1.1. IDENTIFICACION DEL PROBLEMA	1
1.2. FORMULACIÓN DEL PROBLEMA	3
1.3. OBJETIVOS GENERAL	3
1.4. OBJETIVOS ESPECÍFICOS	3
1.5. ALCANCE	3
1.6. JUSTIFICACIÓN E IMPORTANCIA	5
1.7. LIMITACIONES	5
1.8. MARCO REFERENCIAL	5
1.8.1. MARCO DE ANTECEDENTES	5
1.8.2. MARCO TEÓRICO	6
1.8.2.1. TEORÍAS DEL APRENDIZAJE	6
1.8.2.2. TEORÍA GENERAL DE SISTEMAS	8
1.8.2.3. TEORÍA DE LA INFORMACIÓN	9
1.8.3. MARCO CONCEPTUAL	10
1.8.3.1. SOFTWARE	10
1.8.3.2. LENGUAJE DE PROGRAMACIÓN	12
1.8.3.3. PROGRAMACIÓN ORIENTADA A OBJETOS	23
1.8.3.4. UML	28
1.8.3.5. ENTORNO DE DESARROLLO	29
1.8.3.6. ARQUITECTURA MODELO VISTA CONTROLADOR	31
1.8.3.7. CONTROL DE VERSIONES	33
1.8.3.8. REPOSITORIO	36
1.8.3.9. MATERIAL EDUCATIVO COMPUTARIZADO	38
1.8.3.10. METODOLOGÍA PARA EL DESARROLLO DEL MEC.	39
1.8.3.11. ANÁLISIS ESTRUCTURAL	46
1.8.3.12. MODELOS ESTRUCTURALES	46
1.8.3.13. MÉTODO DE LOS ELEMENTOS FINITOS	46
MATERIALES Y MÉTODOS	58
2.1. METODOLOGÍA DE LA INVESTIGACIÓN	58
2.1.1 TIPO DE INVESTIGACION	58
2.1.2 DISEÑO DE INVESTIGACIÓN	58
2.1.3 TECNICAS E INSTRUMENTOS DE RECOLECCION DE DATOS	58
2.1.4. POBLACIÓN Y MUESTRA	59
2.2. DISEÑO METODOLÓGICO DE LA PROPUESTA	59
2.3. HIPÓTESIS	61

2.4.	VARIABLES.....	61
2.5.	FACTIBILIDAD TÉCNICA, OPERATIVA, ECONÓMICA Y PRESUPUESTO DE LA INVESTIGACIÓN.....	63
2.6.	RESULTADOS DEL DIAGNÓSTICO.....	65
2.7.	PROPUESTA DE DESARROLLO.....	71
2.8.	PLAN DE ACTIVIDADES.....	72
2.9.	METODOLOGÍA PARA EL DESARROLLO DE LAS ACTIVIDADES	73
	FASE I: ANÁLISIS DE LAS NECESIDADES EDUCATIVAS	74
	FASE II: DISEÑO DEL MATERIAL EDUCATIVO	80
	ENTORNO PARA EL DISEÑO DEL MEC.	80
	DISEÑO DEL SISTEMA DE COMUNICACIÓN	82
	DISEÑO COMPUTACIONAL	84
	<i>Modelado de casos de uso</i>	85
	<i>Diagramas de secuencia</i>	87
	<i>Diagramas de estado</i>	91
	<i>Diagrama de actividad</i>	92
	<i>Diagrama de despliegue</i>	93
	<i>Diagrama de clases</i>	94
	FASE III: DESARROLLO DE APLICACIÓN.....	96
	FASE VII: PRUEBA PILOTO DEL MATERIAL EDUCATIVO.	109
	RESULTADOS Y DISCUSIÓN	110
	CONCLUSIONES Y RECOMENDACIONES	119
	REFERENCIAS BIBLIOGRÁFICAS	121

INTRODUCCIÓN

Es evidente para quien quiera verlo, que la ingeniería ha tenido y tiene un enorme impacto en la calidad de vida de las sociedades modernas. En los grandes o pequeños proyectos de ingeniería nacientes de las necesidades de los clientes o de la iniciativa por tener una mayor calidad de vida, de las cuales para la realización exitosa de los mismos requiere de muchos estudios y análisis previos, y en varios de los casos requiere del aporte de muchas de las ramas de ciencia.

En esta investigación, se tendrá la oportunidad de involucrar a la ciencia de la ingeniería en tecnologías como base en el desarrollo de una herramienta (software) educativa para la ayuda en el aprendizaje de un aspecto de la ingeniería civil correspondiente al análisis estructural por el método de elementos finitos.

El gran avance en cuanto a la facilidad de uso de las tecnologías de la información ha permitido en la actualidad ayudar a los científicos, ingenieros y/o estudiantes a procesar datos a mayor velocidad, hacer cálculos precisos, realizar consultas de información y por supuesto, a validar los resultados. Además, los mismos científicos en ocasiones requieren crear sus propios programas para probar o simular modelos matemáticos para sus investigaciones.

1.1. IDENTIFICACIÓN DEL PROBLEMA

La educación de hoy procura encontrar a través de los medios tecnológicos el camino para hacer más accesible el conocimiento, sin importar donde se origine éste.

El uso de las tecnologías de información favorece especialmente el desarrollo de forma continua, creando entornos virtuales de aprendizaje, lo que ayuda a aumentar el interés por las asignaturas, la confianza del alumno y el estímulo de la creatividad. Cada vez más entidades educativas, se interesan por las tecnologías y en muchas de las mismas han sustituido los libros por bibliotecas virtuales u otras herramientas digitales, que son empleados como elementos de aprendizaje.

A nivel de educación superior la Universidad Cesar Vallejo – sede Chiclayo, cuenta con modernos laboratorios, además posee excelentes condiciones de acceso a bibliotecas y campus virtual, y a pesar de que la tecnología últimamente es accesible por el alumnado de pregrado; sin embargo existen pocas herramientas alternativas como software que apoye en el aprendizaje en materias de especialidad, específicamente que permitan dar a entender el proceso de análisis estructural por el método de los elementos finitos.

El método de los elementos finitos (MEF en castellano o FEM en inglés) es un método de cálculo utilizado en diversos problemas de ingeniería, que se basa en considerar al cuerpo o estructura dividido en elementos discretos, con determinadas condiciones de vínculo entre sí, generándose un sistema de ecuaciones que se resuelve numéricamente y proporciona el estado de tensiones y deformaciones. También se utiliza en matemáticas como método nodal aproximado para resolver ecuaciones diferenciales en forma numérica.

Es un procedimiento numérico aplicable a un gran número de problemas con condiciones de borde impuestas (en las estructuras las condiciones de borde serían: restricciones y cargas externas). Varios de estos problemas no tienen solución analítica o es muy difícil obtenerla, por lo que se convierte en la única alternativa de resolución. Con este método se pueden resolver sistemas los cuales no son fáciles de resolver mediante modelos matemáticos simples.

El MEF se ha convertido en un sistema apropiado de análisis de estructuras y ha determinado una base muy conveniente para el desarrollo de muchos programas de computación con excelentes resultados y sobre todo para esos casos complejos que requieren de la virtud del cálculo por computadora.

La dificultad que presenta el MEF para la ingeniería, es por lo tedioso que puede ser el uso de este método para la solución de determinados problemas de la ingeniería, sobre todo en el aprendizaje del mismo; y si bien se cuenta con la existencia de software que aplica el método para la obtención de resultados, no lo existen con fines de aprendizaje.

En este orden de ideas, se destaca contar con material educativo computarizado con un conjunto de recursos informáticos diseñados, que permitan que el usuario resuelva diversos ejercicios y observe los procedimientos que llevan a la solución de éstos, obteniendo al mismo tiempo retroalimentación del tema.

1.2. FORMULACIÓN DEL PROBLEMA

¿En qué medida el diseño y desarrollo de material educativo computarizado acerca del método de elementos finitos para el análisis estructural, permitirá disminuir la escasez de herramientas tecnológicas que incidan en la mejora del aprendizaje, así como disminuir las deficiencias en el uso de este método para el alumnado de pregrado de la Escuela de Ingeniería Civil de la Universidad Cesar Vallejo sede Chiclayo?

1.3. OBJETIVOS GENERAL

La presente investigación pretende el diseño y desarrollo de un material educativo computarizado como instrumento de aprendizaje del método de elementos finitos para análisis estructural en los alumnos de pre grado de la Universidad César Vallejo sede Chiclayo, con respecto a un marco referencial que integre una síntesis de planteamientos teóricos y prácticos acerca de este método.

1.4. OBJETIVOS ESPECÍFICOS

- Establecer las características principales con respecto al área de atención sobre al análisis estructural por el método de elementos finitos, que se debe contemplar en el software.
- Analizar y diseñar el software para producir un modelo a implementar, de acuerdo a las necesidades de la investigación.
- Seleccionar las herramientas tecnológicas para el desarrollo mismo del sistema en términos factibles para el cumplimiento de objetivos en el proyecto.
- Realizar las pruebas para lo cual se propuso como material educativo computarizado para el aprendizaje del método de elementos finitos en el análisis estructural.

1.5. ALCANCE

En relación al desarrollo del proyecto se busca implementar un material educativo computarizado que contribuya al proceso de enseñanza – aprendizaje acerca del método de elementos finitos para el análisis estructural.

La práctica de este método radica en que es habitual su uso por los diversos programas de ordenador a la resolución de diferentes estructuras, a diferencia de otros métodos limitados a la tipología de una estructura particular, y es por ello que es de interés a través de este proyecto conlleve al estudiante a comprender a cabalidad las diversas etapas fundamentales del método desde la manera en describir el planteamiento de las operaciones paso a paso hasta la interpretación final de la solución que nos conduce la solución de un problema.

Es de notar, que dicho software educativo pretende que el estudiante adquiera los siguientes conocimientos:

- 1) Identificación estructural
- 2) Discretizar (dividir) la estructura en elementos finitos de geometría simple (elemento) cuyas relaciones entre emplazamientos en sus extremos y las fuerzas inducidas sea conocida.
- 3) Grados de libertad con sus desplazamientos conocidos y desconocidos.
- 4) Matriz de rigidez de cada elemento finito y su vector de cargas nodales equivalentes.
- 5) Matriz de rigidez global y del vector de cargas global de la estructura.
- 6) Solución del sistema de ecuaciones lineales generados de la indeterminación de los desplazamientos desconocidos
- 7) Cálculo de las fuerzas asociadas a los grados de libertad.

Dicho software además de mostrarnos el proceso correcto de uso de este método nos permite analizar, evaluar y validar resultados ante cualquier estructura formadas por barras y vigas.

1.6. JUSTIFICACIÓN E IMPORTANCIA

Este proyecto de Investigación Aplicada, pretende terminar con el desarrollo de un material educativo que permita a los estudiantes de pregrado de la Universidad Cesar Vallejo acerca comprender mejor el análisis estructural por el método de elementos finitos.

Con el avance de la tecnología nos convoca a llevar las necesidades académicas de quienes pretenden enseñar y/o aprender, a la construcción y/o aplicación de nuevos software de ingeniería, que se adapten a las exigencias de la modernidad actual puesto que, al contar con sistemas modernos, el servicio educativo será más eficiente.

1.7. LIMITACIONES

- Se llevará a cabo en la sede de Chiclayo de la Universidad Cesar Vallejo.
- Se le dedicará Quince Horas Semanales

1.8. MARCO REFERENCIAL

1.8.1. MARCO DE ANTECEDENTES

En la educación universitaria venezolana varias instituciones, como la Universidad de Los Andes-Táchira, han desarrollado diversas aplicaciones computarizadas en el área de la Informática Educativa; que van desde libros electrónicos hasta sistemas de instalación y operación en Internet de los contenidos programáticos de algunas asignaturas (González, 2004).

Profesores de la Universidad del Valle de México, campo Villa Hermosa, crearon un software educativo multimedia para la enseñanza de la filosofía y la metodología de la ciencia y la investigación, ante la necesidad de motivar a los estudiantes en la profundización de la lectura de textos densos y complejos, para favorecer la capacidad de reflexión y análisis en esta área, que proporciona las bases para entender, comprender y posteriormente practicar la investigación científica, tan importante en cualquier ambiente, especialmente el universitario (Guzmán y Pérez, 2005).

1.8.2. MARCO TEÓRICO

1.8.2.1. TEORÍAS DEL APRENDIZAJE¹

Todas las aproximaciones psicológicas al fenómeno del aprendizaje humano tienen algo que decir como fundamento para el diseño de ambientes de enseñanza - aprendizaje. Sin embargo, los aportes no necesariamente son convergentes, como no lo es la perspectiva desde la cual se analiza el fenómeno en cada caso, ni los métodos usados para obtener el conocimiento. Si hubiera una teoría que atendiera todos los aspectos del fenómeno, que abarcara las demás teorías, no habría que estudiar las otras; pero la realidad es diferente, de ahí la necesidad de por lo menos conocer los puntos más importantes de los diferentes aportes en relación al tema.

Las aproximaciones al fenómeno del aprendizaje oscilan entre dos polos: conductismo y cognoscitivismo.

En el primer polo no se toma en cuenta el organismo (el sujeto que aprende), sólo las condiciones externas que favorecen su aprendizaje; por esto se habla de un modelo de "caja negra" en el que lo fundamental es la programación en pequeños pasos, de eventos que conduzcan a lograr el resultado esperado (respuesta) y el reforzamiento de las respuestas, que confluyen hacia el logro de lo que se desea.

En el otro polo lo que cuenta es el individuo, el aprendiz, con todo su campo vital, su estructura cognoscitiva, las expectativas que tiene. Por contraposición se habla de un modelo de "caja traslúcida" en el que lo que cuenta es el aprendiz dentro de su entorno psicológico y social. La motivación interna, la significancia, el procesamiento de la información, las aptitudes de las personas, entre otros, son tomados en cuenta como factores que promueven el aprendizaje.

En medio de los enfoques anteriores, es posible encontrar una combinación de ambos. En el cual se comparten algunas características. Ya no se habla de caja negra como en el primer enfoque, pero tampoco se considera al aprendiz, como único elemento de relevancia (enfoque de "caja traslúcida").

¹ Universidad de Concepción, Chile. Departamento de Ingeniería Informática y Ciencias de la Computación. Revista Ingeniería Informática. Edición número 6. 15 Septiembre 2000. "Ingeniería de software educativo, teorías y metodologías que la sustentan". Pedro Salcedo. [online]. Consultado en: <<http://www.inf.udec.cl/~revista/>> el 26 de Abril de 2010.

(Bertoglio, 2004)

A pesar de las anteriores diferencias, las teorías de aprendizaje tienen en común su objeto de estudio: el aprendizaje. No es de extrañar, por consiguiente, que se logre un efecto de "triangulación" (ver de varios ángulos un mismo asunto) cuando se analizan los distintos aportes. Desde cada teoría existe una perspectiva que complementa a otras.

El evolucionismo genético de Piaget, propone que el aprendizaje está subordinado al desarrollo orgánico y de estructuras cognoscitivas (a la madurez del estudiante), además se basa en la experiencia, actividad inquisitiva sobre el objeto de conocimiento. Por otra parte para Piaget el aprendizaje resulta de alcanzar nuevos estados de equilibrio a partir de disequilibrios cognitivos que se solucionan mediante asimilación de nuevos conocimientos y acomodación de las estructuras cognoscitivas a partir de experiencias.

"El uso instruccional del computador reproduce los viejos esquemas de la instrucción programada en cualquiera de sus modalidades. Por el tipo de postulados epistemológicos de su ciencia base, le es imposible a esta tecnología trazarse como objetivo desarrollar la inteligencia. El uso interactivo del computador hace parte de una tecnología educativa derivada del estructuralismo genético de Piaget. Esta ciencia base ha desarrollado los instrumentos conceptuales y epistemológicos necesarios para darle significado empírico a enunciados acerca del desarrollo de la inteligencia."²

Una teoría pedagógica de gran Dauge actualmente y que parte de las teorías de Piaget es el "construccionismo", cuyos principios han sido elaborados por Papert desde la implementación del lenguaje Logo, al final de la década de los 60's.

Esta teoría constructivista plantea un cambio a la educación tradicional, de modo que todo desarrollo de herramientas para asistir el aprendizaje, debieran tener sus bases en esta teoría. Papert se refiere al construccionismo como una teoría que maximiza lo aprendido y minimiza lo enseñado.

Al fracaso de la educación tradicional se han atribuido muchas razones: el curriculum no tiene relación con la vida real de los alumnos, los docentes no están bien preparados, los estudiantes carecen de conocimientos básicos, falta individualización y muchas otras. El cambio que requiere con urgencia la

² "El uso instruccional vs. el uso interactivo del computador respecto al desarrollo de la inteligencia", Hernan Escobedo David. Aparece en Memorias del Primer Congreso Colombiano de Informática, Educación y Capacitación. 1987.

educación no se dará apoyando las labores usuales de la educación (tareas administrativas, empleándolo para hacer trabajos, como enciclopedia, como calculadora, o simplemente asistiendo la evaluación tradicional), sino cambiando sus cimientos y volviendo a pensar la pedagogía. Hoy, el reto de la educación es aprovechar los nuevos medios (vídeo, audio, computadores, inteligencia artificial, realidad virtual, etc.) para atraer al estudiante y permitirle la construcción de su conocimiento.

1.8.2.2. TEORÍA GENERAL DE SISTEMAS³

Los objetivos de la teoría general de sistemas pueden ser fijados a diferentes grados de ambición y de confianza. A un nivel de ambición bajo, pero con un alto grado de confianza, su propósito es descubrir las similitudes en las construcciones teóricas de las diferentes disciplinas, cuando éstas existen, y desarrollar modelos teóricos que tengan aplicación al menos en dos campos diferentes de estudio. A un nivel más alto de ambición, pero quizás, con un grado de confianza menor, espera desarrollar algo parecido a un “espectro” de teorías, un sistema de sistemas que pueda llevar a cabo la función de un gestalt⁴ en las construcciones teóricas. Este espectro o gestalt ha tenido gran valor en campos específicos del conocimiento humano, al dirigir las investigaciones hacia los vacíos que ellos revelan.

El conocimiento no es algo que exista y crezca en abstracto. Es una función del organismo humano y de las organizaciones sociales. El conocimiento crece a través de la recepción de información, es decir, de la obtención de mensajes capaces de reorganizar el conocimiento receptor. Por lo tanto, el crecimiento del conocimiento, en general, depende directamente de este flujo de comunicaciones entre científicos.

Mientras más se divide la ciencia en subgrupos y menos sea la comunicación entre las disciplinas, mayor es la probabilidad de que el crecimiento total del conocimiento sea reducido por la pérdida de comunicación relevante. El esparcimiento de la sordera especializada significa que una persona que debiera saber algo que otra conoce es incapaz de encontrarlo por la falta de un “oído generalizado”. Ahora bien, uno de los principales objetivos de la teoría general de sistemas es la multiplicación de estos oídos generalizados

³ “Introducción a la teoría general de sistemas”. Johansen Oscar. Bertoglio, Mexico. Limusa 2004, pág. 21-23

⁴ Gestalt: (palabra alemana que significa, aproximadamente “configuración”). Es la experiencia perceptiva normal en la cual la totalidad es vista o comprendida como algo más que la simple suma de sus partes...El ejemplo clásico es un dibujo que puede ser percibido ya sea como florero o como dos caras humanas.

J.A. Brusel y G.L. Cantzlaar: Diccionario de (Miller) (Peña, 2002) (Arias, 2006) Psiquiatría (México, CECSA Ed 1972) p.128.

y el desarrollo de un marco referencial de teoría general que permita que un especialista pueda alcanzar captar y comprender la comunicación relevante de otro especialista.

1.8.2.3. TEORÍA DE LA INFORMACIÓN⁵

Esta introduce el concepto de información como una cantidad mensurable, mediante una expresión isomórfica con la entropía negativa en física. En efecto, los matemáticos que han desarrollado esta teoría han llegado a la sorprendente conclusión de que la fórmula de la información es exactamente igual a la fórmula de la entropía, solo con el signo cambiado, de donde se deduce que:

$$\text{Información} = -\text{entropía}$$

Ahora bien, la entropía (positiva en física es una medida de desorden). Luego la información (o entropía negativa) es una medida de organización. En este sentido es interesante observar una conclusión a que ha llegado J.J. Miller⁶ que señala que, mientras más complejos son los sistemas (entendiéndose por complejidad el número posible de estados que puede presentar cada parte y el número de las posibles relaciones entre esas partes) mayor es la energía que dichos sistemas destinan tanto a la obtención de la información como a su procesamiento, decisión, almacenaje y/o comunicación.

⁵ "Introducción a la teoría general de sistemas". Johansen Oscar. Bertoglio, Mexico. Limusa 2004, pag 29-30

⁶ J.J. Miller. "The Living Systems", Behavioral Sciences.

1.8.3. MARCO CONCEPTUAL

1.8.3.1. SOFTWARE⁷

En primera instancia, el término inglés software, que corresponde a soporte lógico o programa en español, es aplicable a toda colección de instrucciones que sirve para que el computador cumpla con una función o realice una tarea.

Generalmente, el software se suele clasificar en tres grandes grupos:

SOFTWARE DE SISTEMA: Esta clase de software incluye a todos aquellos programas que le permiten al usuario establecer una interacción con el sistema para de esta forma lograr controlar al hardware. Además de esto, este software se encarga de brindarle soporte a diferentes programas. Gracias al software de sistema, el usuario no debe estar al tanto de toda la información del sistema informático, vinculada con los discos, memoria, dispositivos, teclados, puertos, impresoras, entre otros.

SOFTWARE DE PROGRAMACIÓN: Son aquellos programas cuya función depende o se encuentra relacionada a la programación informática a través de distintos lenguajes de programación.

SOFTWARE DE APLICACIÓN: El software de aplicación permite realizar tareas puntuales como aquellas que implican hojas de cálculo, procesadores de texto u otros sumamente específicos, como los programas educativos, de contabilidad o para componer música. También puede incluir videojuegos, de cálculo numérico o para diseño.

Dentro del software de aplicación, también podemos hacer una mención especial sobre el:

SOFTWARE EDUCATIVO: El software educativo es todo aquel soporte inspirado para la enseñanza y el aprendizaje de quien lo utilice. Es un tipo pedagógico moderno, y está directamente relacionado a la proliferación de las computadoras en los últimos años, de fuerte raigambre en los colegios, los hogares y las bibliotecas. Existen distintos tipos de software educativo, como son:

- Software educativo de sistema tutorial
- Software educativo de ejercitación y práctica
- Software educativo de simuladores y juegos

7 <http://www.tipos.co/tipos-de-software/>
"SISTEMA DE ANÁLISIS ESTRUCTURAL EN PLATAFORMA
JAVA PARA VIGAS CONTINUAS POR EL MÉTODO DE
ELEMENTOS FINITOS"

Si nos atenemos a su licencia; muchas de ellas pueden prestarse a confusión, por eso se aclarará lo que significan los diferentes calificativos aplicados al software.

- **Software comercial**

Es el desarrollado por una empresa con intención de venderlo y obtener unos beneficios. No debe confundirse con propietario, puesto que hay software libre que es comercial, aunque es cierto que la mayoría del software comercial es propietario.

- **Software semilibre**

Es aquel que mantiene las mismas características que el software libre para los usuarios individuales, entidades educativas o sin ánimo de lucro, sin embargo, prohíbe esas libertades para su uso comercial o empresarial.

Por ejemplo, el software de dominio público significa que no está protegido por el copyright, por lo tanto, podrían generarse versiones no libres del mismo, en cambio el software libre protegido con copyleft impide a los redistribuidores incluir algún tipo de restricción a las libertades propias del software así concebido, es decir, garantiza que las modificaciones seguirán siendo software libre.

También es conveniente no confundir el software libre con el software gratuito, éste no cuesta nada, hecho que no lo convierte en software libre, porque no es una cuestión de precio, sino de libertad.

- **Software propietario**

Es aquel que no es libre ni semilibre; por lo tanto, su redistribución, modificación y copia están prohibidas o, al menos, tan restringidas que es imposible hacerlas efectivas.

- **Freeware**

No tiene una definición clara y precisa, sin embargo, suele usarse para clasificar al software que puede redistribuirse libremente pero no modificarse, entre otras cosas, porque no está disponible su código fuente. El freeware no es software libre.

- **Shareware**

Es un software que permite su redistribución, sin embargo, no viene acompañado de su código fuente y, por tanto, no puede ser modificado.

1.8.3.2. LENGUAJE DE PROGRAMACIÓN

En general, se puede clasificar a los lenguajes de programación en tres categorías:

- 1) Lenguajes compilados
- 2) Lenguajes interpretados
- 3) Híbridos entre los dos anteriores

En la primera categoría, tenemos a lenguajes como Fortran o C/C++. Estos lenguajes se distinguen porque usan un compilador que traduce el código de alto nivel a código máquina. Este código máquina generado está en el ejecutable (en Windows se distingue por la extensión .exe) y es dependiente de la máquina para la que se compiló. Es decir, si queremos correr un programa en un lenguaje compilado en varios sistemas, debemos compilar para cada uno de ellos. La principal ventaja de estos lenguajes es que una vez que se tiene el ejecutable este trabaja directamente con lenguaje máquina, por lo que la ejecución en general es más eficiente.

En la segunda categoría tenemos a los lenguajes interpretados, los cuales son compilados en tiempo de ejecución línea por línea. Son lenguajes en los que al ejecutar un programa cada línea es compilada y luego ejecutada. Ejemplos de lenguajes interpretados son Smalltalk, JavaScript y PHP. Estos lenguajes usan un intérprete que va ejecutando los programas línea por línea. La ventaja de esto es que el programa se vuelve independiente de la plataforma, por lo que puede correr en cualquier sistema que tenga un intérprete para el lenguaje en el que está escrito el programa. Sin embargo, estos lenguajes son en general menos eficientes, pues cada instrucción tiene que ser interpretada antes de ejecutarse.

La tercera categoría es cuando se utilizan técnicas para hacer más eficientes los lenguajes interpretados. La mayoría de los lenguajes catalogados como interpretados usan ahora estas técnicas para solucionar el problema de la eficiencia. Generalmente lo que hacen es combinar la compilación con la interpretación. Por ejemplo, Java traduce el código a un código intermedio (bytecode) que interpreta la Máquina Virtual de Java (JVM), pero también usa la Compilación Justo a Tiempo (Just-in-time Compilation o JIT Compilation) para pasar a código máquina secciones de código que se usan regularmente. Los lenguajes

de .NET son otro ejemplo; en dichos lenguajes se compila a código intermedio (Common Intermediate Language o CIL) que el intérprete CLR (Common Language Runtime) compila en tiempo de ejecución.

La siguiente tabla 1 muestra los lenguajes de programación considerados para hacer este proyecto.

Lenguaje de programación	Descripción breve	Características
Fortran	Fue desarrollado por IBM para usarse en aplicaciones científicas y de ingeniería con cálculos matemáticos complejos.	Optimizado para realizar operaciones sobre arreglos Popular en el ámbito científico Permite cómputo en paralelo Múltiples bibliotecas para uso científico
C	Popular por ser el lenguaje de desarrollo del sistema operativo UNIX. Se usa generalmente para programar los sistemas operativos de propósito general.	Lenguaje estructurado Eficiencia en cálculos al ser un lenguaje compilado Permite manipulación directa de la memoria con el uso de apuntadores Existen muchas bibliotecas de funciones en este lenguaje
Java	Lenguaje de programación orientado a objetos y multiplataforma	Programación Orientada a Objetos Multiplataforma Seguro y Robusto
Visual C#	Es uno de los lenguajes desarrollados por Microsoft para facilitar el desarrollo de aplicaciones para Windows	Programación Orientada a Objetos <ul style="list-style-type: none"> • Usa Visual Studio como su IDE

Tabla 1: Lenguajes de programación

La principal deficiencia del programa en Fortran es la falta de una interfaz gráfica amigable para el usuario. Es por ello que para la elección del lenguaje de programación se tomará en cuenta principalmente este punto. La carencia de interfaces gráficas en los programas de uso científico es muy común. Los científicos en su mayoría programan en lenguajes como C/C++ o Fortran para realizar programas que los apoyen en sus investigaciones. Estos lenguajes son los preferidos en el ámbito científico debido a que son lenguajes compilados, por lo que son más eficientes para programas que realizan muchas operaciones matemáticas.

Existen bibliotecas gráficas en lenguajes como C y Fortran que podrían permitir a los científicos dar mayor vistosidad a los programas con el uso de interfaces gráficas. Sin embargo, hay poca información sobre la implementación de interfaces con estos lenguajes, sus interfaces no son tan atractivas gráficamente y son más complicadas de implementar. Es por eso que estos lenguajes se descartan para la implementación de la interfaz gráfica.

En cuanto a los dos restantes, ambos son lenguajes híbridos, por lo que no son tan eficientes como los lenguajes compilados. Sin embargo, estos lenguajes tienen la ventaja de estar basados en el paradigma de Programación Orientada a Objetos (POO). Ésta es una gran ventaja pues la POO ayuda al programa a ser fácil de mantener y de hacer modificaciones. Además, debido a la popularidad de estos lenguajes, se cuenta con entornos de desarrollo muy potentes que pueden ser usados para acelerar aún más el proceso de la creación de interfaces gráficas.

Visual C# es un lenguaje creado por Microsoft. Esto le da ventajas y desventajas. Debido a que Windows es el sistema operativo más usado, las aplicaciones desarrolladas para este sistema operativo tienen un gran alcance de usuarios. Pero el problema es que no se pueden correr programas propios de Windows, por ejemplo, los creados usando Visual C#, en otros sistemas operativos, como MacOS o Linux. Debido a que se pretende que el programa tenga un número grande de usuarios, se debe tomar en cuenta el hecho de que la aplicación debe correr en el mayor número de máquinas posible. Visual C# nos limitaría a máquinas con Windows, por lo que se descarta como opción.

El único que falta analizar y que será el elegido para la aplicación de vulcanología, es Java. Este lenguaje, que como se mencionó tiene el paradigma de programación orientada a objetos, además tiene la ventaja de ser multiplataforma. Un programa escrito en Java puede ser ejecutado en cualquier computadora,

siempre y cuando ésta tenga instalada la Máquina Virtual de Java. Esto le da a Java un gran alcance, pues al menos para los sistemas operativos más populares (Windows, MacOS y las variantes de Linux) existe una implementación de la Máquina Virtual de Java. Es por esto que se decide usar este lenguaje de programación, así como por otras características de las que se hablará más adelante.

LENGUAJE DE PROGRAMACIÓN JAVA

El lenguaje de programación Java fue liberado por Sun en 1995. Es un lenguaje de programación concurrente, basado en clases y orientado a objetos. Fue creado con el lema “write once, run anywhere” (WORA), es decir, está pensado para que los programas corran en cualquier máquina sin la necesidad de compilar para cada una de ellas. Para esto se usa la Java Virtual Machine (JVM) que tiene que estar instalada en la máquina en donde se desee correr un programa de Java.

A continuación se muestran algunas estadísticas que publica Oracle en la página oficial de Java (<http://www.java.com/es/about>)

- El 97% de los escritorios empresariales ejecutan Java
- El 89% de los escritorios (o computadoras) en Estados Unidos ejecutan Java
- 9 millones de desarrolladores de Java en todo el mundo
- La primera opción para los desarrolladores
- La primera plataforma de desarrollo
- Mil millones de teléfonos móviles ejecutan Java
- El 100% de los reproductores de Blu-ray incluyen Java
- 5 mil millones de Java Cards en uso
- 125 millones de dispositivos de televisión ejecutan Java
- 5 de los 5 principales fabricantes de equipos originales utilizan Java ME

El proyecto del lenguaje Java fue iniciado en 1991 por James Gosling. Mike Sheridan y Patrick Naughton. Su nombre inicial fue Oak, pero al final fue llamado Java, nombre tomado del café al que muy a menudo iban los creadores del lenguaje. La primera versión liberada fue Java 1.0 en 1995. En 1998 se liberó la versión de Java 2, a partir de la cual ya había versiones diferentes para diferentes tipos de plataformas:

J2SE (Standard Edition), J2EE (Enterprise Edition) y J2ME (Micro Edition). Entre 2006 y 2007 Sun liberó la mayor parte del código de Java usando la licencia GNU GLP (General Public License) con lo que lo hizo Free and Open Source Software (FOSS). Entre el 2009 y el 2010 Oracle adquirió la compañía de Sun Microsystems con lo que adquirió entre otras tecnologías a Java.

VERSIONES DE JAVA

La tabla 2 muestra las diferentes versiones de Java con una breve descripción de sus características más importantes.

FECHA DE LIBERACIÓN	VERSIÓN	DESCRIPCIÓN
1995	JDK Alfa y Beta	Fueron las primeras versiones públicas de Java que se usaron para depurar, por lo que eran aún muy inestables
23 de enero de 1996	JDK 1.0	Originalmente llamado Oak, fue la primera versión estable de Java.
19 de febrero de 1997	JDK 1.1	
8 de diciembre de 1998	J2SE 1.2	A partir de esta versión se le llama a la versión como Java 2 Standard Edition (J2SE) y se continua con este estándar hasta la versión J2SE 5.0. Las mejoras más importantes fueron: La API de Swing se integró al lenguaje El compilador JIT se integró a la JVM por primera vez.

		<p>Se agrega Java IDL (Interface Definition Language) que es una implementación de CORBA en Java.</p> <p>Se agrega JCF (Java Collections Framework) que es un conjunto de clases e interfaces que implementan colecciones reutilizables de estructuras de datos (Listas, Pilas, Colas, etcétera)</p>
8 de mayo de 2000	J2SE 1.3	<p>Las mejoras más importantes fueron:</p> <p>Hotspot incluido en JVM</p> <p>RMI modificado para soportar compatibilidad con CORBA</p> <p>Se incluye JNDI (Java Naming and Directory Interface)</p> <p>Se incluye JPDA (Java Platform Debugger Architecture)</p> <p>Se incluye JavaSound</p>
6 de febrero de 2002	J2SE 1.4	<p>Fue la primera versión desarrollada a cargo del Java Community Process. Las mejoras que incluyó fueron:</p> <p>mejora en expresiones regulares</p> <p>permite encadenamiento de excepciones</p> <p>soporte para IPv6</p> <p>permite entrada y salida con non-blocking</p> <p>se agrega API para E/S de imágenes en JPEG y PNG</p> <p>se integra parser de XML</p>

		<p>Se incluyen extensión de seguridad y criptografía (JCE, JSSE, JAAS)</p> <p>Se incluye Java Web Start</p>
30 de septiembre de 2004	J2SE 5.0	<p>Se cambia el formato de numeración de las versiones. Se agregó lo siguiente:</p> <p>Generics: eliminan necesidad de la mayoría de casts</p> <p>Metadata: se pueden agregar etiquetas con datos adicionales para ser procesados como metadata</p> <p>Autoboxing: conversión automática entre tipos primitivos y sus clases respectivas (por ejemplo, convertir int en Integer)</p> <p>Enumeraciones: con la palabra clave enum se puede definir listas de valores sin necesidad de usar constantes para mayor claridad en código</p> <p>Varargs: se ponen con tres puntos seguidos (...) y permiten mandar a una función una cantidad variable de parámetros</p> <p>For each: es una sentencia for mejorada para iterar sobre elementos de un arreglo (o de un objeto Iterable) que no van a ser modificados.</p>
11 de diciembre de 2006	Java SE 6	<p>Se quita el .0 de el número de versión. Se agregó lo siguiente:</p> <p>Soporte para versiones viejas de Windows</p> <p>Soporte para Scripting Languages</p> <p>Mejoras de rendimiento en Swing</p>

		Se mejora la funcionalidad de Web Service
2007- 2014	Java 6 updates	A esta versión se le dio soporte durante varios años sin hacer cambios en la API. Estas modificaciones se enfocaron en: Actualizaciones de seguridad Mejorar la usabilidad para el usuario final
28 de julio de 2011	Java SE 7	Las mejoras de esta versión son: Se pueden usar Strings en switch Manejo de recursos en sentencias try Catch de excepciones múltiples Utilidades para la concurrencia Biblioteca de E/S para mejorar la independencia de plataforma. Se mejora Java 2D para aprovechar las capacidades de las nuevas GPUs
2011-2014	Java 7 updates	Las modificaciones se enfocaron en: Actualizaciones de seguridad Mejorar la usabilidad para el usuario final Corrección de bugs
18 de marzo de 2014	Java SE 8	Se permite a los programadores agregar código de JavaScript en las aplicaciones de Java Lambda expressions: permiten usar el código como datos. Mejoras de rendimiento en HashMaps Mejoras de seguridad Mejoras en la gestión de fechas y tiempo.

	Java 9	<p>nueva versión que mejora la anterior (en difusión)</p> <p>Se agrega paquete para usar JavaBeans (componentes reutilizables de Java)</p> <ul style="list-style-type: none"> • Se agrega JDBC (acceso a base de datos) • Se agrega RMI (Remote Method Invocation, para programar arquitecturas cliente-servidor) • <p>Se agrega compilador JIT (just-in-time) en plataformas de Windows, lo que mejora el rendimiento de los programas.</p>
--	--------	--

Tabla 2: Diferentes versiones de Java

LA API DE JAVA

Una API (Application Programming Interface) define cómo van a interactuar los componentes de software entre ellos y cuenta con un conjunto de funciones y procedimientos que el usuario del lenguaje puede usar. En Java, la API viene organizada en paquetes, dentro de los cuales se encuentran las clases que pertenecen a ese paquete. En otros lenguajes, generalmente se usan bibliotecas de funciones para definir la API por lo que si se quiere usar una biblioteca, se incluye en el programa (en C por ejemplo se usa la sentencia include). Pero como Java es un lenguaje orientado a objetos, a API se define a través de clases, que si se quieren usar deben ser importadas al programa usando la sentencia import.

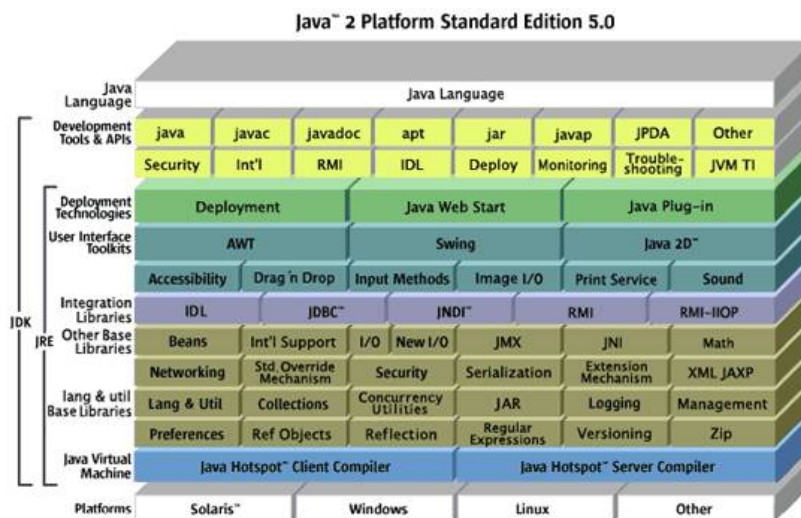


Ilustración 1: API de Java

La ilustración 1 muestra la multitud de paquetes que contiene la API de Java. Como se puede observar existen paquetes para propósitos muy específicos. Para este proyecto son especialmente importantes los que se muestran en la categoría User Interface Toolkits (herramientas para interfaz de usuario).

MÁQUINA VIRTUAL DE JAVA

La máquina virtual de java es la tecnología responsable de la independencia que tiene Java con respecto al hardware y al sistema operativo en el que corre, el tamaño pequeño de su código compilado y la habilidad para ofrecer seguridad a los usuarios contra programas maliciosos. Se trata de una computadora virtual, que al igual que una computadora física tiene un set de instrucciones y manipula la memoria para correr programas.

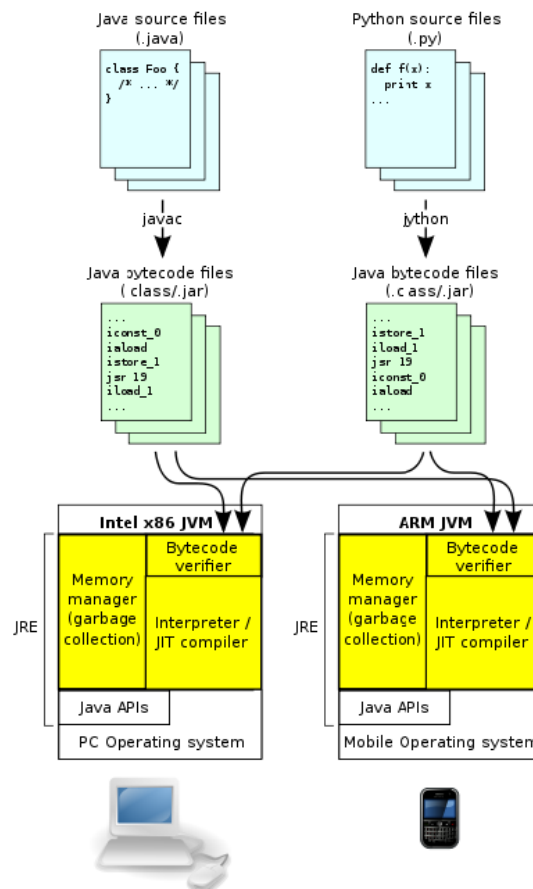


Ilustración 2: Arquitectura JVM

La Máquina Virtual de Java no entiende el lenguaje de Java; lo que entiende es el formato binario del tipo de archivo con extensión .class, el cual contiene el bytecode del programa. El bytecode es un código intermedio que la JVM compila justo antes de ejecutarse con el compilador JIT (Ilustración2: Arquitectura JVM).

Aunque al programar en Java nunca nos tendremos que preocupar por todos estos detalles sobre la JVM, es importante conocerlos para entender cómo trabaja. Es principalmente gracias a la JVM que Java tiene tantas ventajas sobre otros lenguajes, como la seguridad, la robustez y ser un lenguaje multiplataforma. Pero también hay otras características que Java tiene, como la facilidad de mantenimiento y la reutilización de código, que no son producto de la JVM, sino del paradigma sobre el que se diseñó Java.

1.8.3.3. PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos, abreviada POO, es un paradigma de programación que usa objetos para representar los datos (atributos en POO) y procedimientos (métodos en POO) de un programa. Se trata de un paradigma que tiene como objetivo permitir el diseño de sistemas de software modulares y reutilizables. El paradigma orientado a objetos está basado en conceptos como la programación estructurada y los tipos abstractos de datos. Un objeto es un tipo abstracto de dato que tiene como agregados el polimorfismo y la herencia. En la programación estructurada se hace una separación entre código y datos, cuando en la programación orientada a objetos ambos conceptos se unen para formar un objeto, el cual tiene un estado (atributos) y un comportamiento (métodos).

Algunas de las ventajas que ofrece la programación orientada a objetos son:

- Permite definir el programa de una manera más natural con el uso de objetos que describen el problema
- Facilitar el mantenimiento del software al hacer programas más claros.
- Facilitar la evolución, extensión o mejoramiento del software por su modularidad. Fomenta la reutilización del código al programarse en módulos que pueden ser usados por otros programas.
- Ahorra tiempo de desarrollo al permitir desarrollar en paralelo módulos independientes del software

Los conceptos en los que se basa este paradigma se empezaron a desarrollar desde la década de 1950, pero fue hasta la década de 1960 cuando el concepto de objeto en programación tomó forma. Simula 67, un lenguaje de programación desarrollado en el Centro de Cómputo Noruego en Oslo, fue el primer lenguaje de programación en aplicar el concepto de objeto. Smalltalk fue desarrollado durante la década de 1970 por Xerox PARC y fue el primero en introducir el término Programación Orientada a Objetos para representar el uso de objetos y mensajes en un programa de computadora. Después de Smalltalk, muchos lenguajes existentes adoptaron la POO como paradigma de programación y se crearon otros lenguajes partiendo de la POO.

Ejemplos de lenguajes que usan la POO son:

- C++
- Objective-C
- Smalltalk
- Java
- C#
- Perl
- Python
- Ruby
- PHP

Algunos conceptos relacionados a la programación orientada a objetos son:

Clase. - Definición de los atributos y el comportamiento de un tipo de objeto.

Instancia. - Se trata de la creación de un objeto a partir de las reglas definidas en la clase.

Atributos. - Son las características del objeto (datos) que se definen en la clase.

Métodos. - Son subrutinas asociadas a un objeto y que se definen en la clase para describir el comportamiento del objeto.

Abstracción. - Es la definición de un objeto en base a sus características esenciales (atributos y comportamiento). La abstracción es una de las características más importantes en el diseño orientado a objetos, pues permite modelar un problema en base a un conjunto de clases.

Modularidad. - Al crear clases, el software se separa naturalmente en módulos, que pueden ser utilizados como parte de varios programas.

Mensajes. - Es la comunicación hacia un objeto que le indica que ejecute un método con los parámetros indicados en el mensaje. Las características básicas que un lenguaje de programación debe tener para entrar en el paradigma orientado a objetos son: encapsulamiento, herencia y polimorfismo.

- Encapsulamiento

El encapsulamiento se refiere a agregar dentro de un objeto todos los recursos necesarios para que el objeto funcione, esto es, agregar los métodos y los atributos que requiere. La idea del encapsulamiento es que el código no inflencie en otro código, ni sea influenciado por otro código. Las clases permiten que los datos y los métodos estén estrechamente ligados, formando una “cápsula” congruente con el objeto que se está definiendo. El encapsulamiento es usado para esconder los valores de los atributos dentro de una clase, de modo que sólo el objeto pueda tener acceso a ellas y no lo tengan otros objetos externos. Para esto los atributos deben ser definidos como privados, es decir, sólo los métodos de la clase pueden acceder a ellos o modificarlos. Para acceder a los atributos desde otras clases se proporcionan en la clase métodos específicos para esto:

Para consultar el valor de un atributo desde una clase externa se tiene que definir un método get para este atributo.

Para modificar el valor de un atributo desde una clase externa se tiene que definir un método set para este atributo.

Un ejemplo puede ayudar mucho para entender cómo funciona esto. Un ejemplo sencillo que puede ser la clase Vehículo. Para la clase Vehículo tenemos que definir sus atributos y sus métodos. Debido a que una clase es una abstracción no es necesario incluir todos los atributos y métodos (comportamientos) que se pueden encontrar en un vehículo real. Tomaremos los que consideremos importantes. Para este caso se elegirán los atributos: posición y color. Para sus métodos escogeremos sólo el método mover. En este caso, para nuestros propósitos estos elementos son suficientes para describir un vehículo. Es importante que cuando usemos la clase vehículo, la usemos sólo a través de sus métodos, para no cambiar directamente sus atributos, pues esto puede generar errores. Por ejemplo, se tiene el método mover, el cual modificará la posición de acuerdo a la implementación que se haya hecho del método. En la clase vehículo consideraremos cómo es que se tendrá que mover o con qué reglas para que la posición se actualice correctamente. Tanto para posición como para color, se pueden agregar métodos get y set para poder modificarlos. La ventaja de usar estos métodos, es que podemos implementar en ellos medidas de prevención para que el usuario de la clase no ponga valores no válidos.

- Herencia

La herencia es una de las características más poderosas de la programación orientada a objetos, pues fomenta la reutilización del código. La herencia se refiere a que se pueden crear clases que son hijas de otra clase. A la clase padre se le llama superclase, la cual contiene atributos y métodos comunes para todas las clases hijas. Las clases hijas se llaman subclases y son las que heredan estos elementos comunes de las superclases, pero contienen también atributos y métodos propios. De esta manera, se puede ahorrar mucho código por ejemplo al implementar métodos una sola vez en la superclase, además de que el código es más fácil de mantener, pues sólo se modifica en una parte y esta modificación pasa automáticamente a las subclases. Es importante mencionar que los métodos heredados por las subclases pueden ser re implementados por ellas para que se adecuen a la subclase en específico.

La superclase directa es aquella que está explícitamente declarada como superclase de la subclase. En Java se utiliza la palabra reservada `extends` para indicar esto. Sin embargo, también existen superclases indirectas, que son aquellas que no aparecen explícitamente en la clase que hereda, pero que aparecerá en alguna clase de la jerarquía de clases. La jerarquía en Java siempre empieza con la clase `Object`, es decir, todas las clases en Java son subclases de la clase `Object`, ya sea directa o indirectamente. En Java sólo se puede heredar directamente de una clase. En el ejemplo con la clase `Vehículo` podríamos crearle dos subclases: la clase `Carro` y la clase `Avión`. En la clase `Vehículo` teníamos los atributos `posición` y `color`, y el método `mover`. Un `carro` y un `avión` comparten estas características, por ello cuando creemos las clases indicaremos que heredan de `Vehículo`. Pero, además, cada uno de ellos puede tener atributos y métodos propios y para el método `mover`, cada uno podría tener su propia implementación, pues un `carro` y un `avión` en la vida real, no se mueven de la misma manera, por lo que en el programa se tendría que tomar esto en consideración. Este último punto sobre la implementación de métodos propios en la subclase, tiene mucho que ver con el concepto de polimorfismo, que se verá a continuación.

- Polimorfismo

El polimorfismo está muy ligado a la herencia. El polimorfismo se refiere a que dentro del código se puede tratar a todas las subclases como si se tratara de su superclase. Esto simplifica mucho la programación y la hace más clara. El enfoque que se da es el de tratar los problemas de manera general y no de manera específica.

Para el ejemplo de la clase Vehículo y las subclases Carro y Avión, esto se podría explicar de la siguiente manera. Por ejemplo, digamos que tenemos un método en nuestro programa principal se recibe un objeto de tipo Vehículo. Aunque el método se define como que recibe objetos del tipo Vehículo, si le mandamos un objeto de tipo Carro o Avión, no fallará, pues al ser subclases de Vehículo, los considera vehículos. Supongamos que dentro del método se manda a llamar al método mover del objeto. En caso de que se trate de un objeto de tipo Carro, se mandará a llamar al método mover de la clase Carro, y si es un objeto Avión, se mandará a llamar al método mover de la clase Avión. Es decir, con la misma llamada del método, se tienen diferentes respuestas. Eso es el polimorfismo. Otra de las formas donde se implementa el polimorfismo es dentro de una misma clase. Para este caso, lo que hacemos es implementar varios métodos con un mismo nombre. Este concepto es llamado sobrecarga de métodos. Cuando un método de este tipo se ejecuta en un programa, Java identifica de cuál se trata por el número de parámetros que cada uno recibe. Entonces por ejemplo, para el método mover de nuestro programa, podríamos tener un método que no reciba nada y otro que reciba la velocidad a la que se va a mover. Ahora queda claro que Java debe en gran medida su éxito a la Programación Orientada a Objetos. Como se comentó en esta sección, las mayores ventajas de este paradigma de programación son brindar claridad al código, de manera que sea más fácil desarrollar y mantener aplicaciones. Éste es un punto muy importante que se tomó en cuenta para la decisión del lenguaje de programación, pues un lenguaje que trabaje con este paradigma logrará que el programa pueda servir por más tiempo, al ser más fácil adaptarlo en caso de requerir cambios. Ahora el punto importante es decidir qué herramientas usar para crear la interfaz gráfica. Para este punto se tendrá que decidir el entorno de desarrollo con el que se va a trabajar, tema que se tratará en la siguiente sección.

1.8.3.4. UML

El estudio profundo de un problema de software requiere la utilización de un plan bien analizado, que organice el proceso de desarrollo de manera que los analistas, clientes, desarrolladores y todos los involucrados lo comprendan y convengan con él. El UML proporciona tal organización (Schmuller, 2002). Por esta causa, se ha convertido en el estándar de facto para modelar software.

UML se puede definir como un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema de software. Ofrece una forma estándar para escribir un plano del sistema (SparxSystemsPtyLtd, 2011). Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas haciendo uso de reglas. Son trece (13) en total y presentan diversas perspectivas del sistema, a las que se conocen como modelo. Entre los diagramas más usados se encuentran los de: clases, casos de uso y secuencia (Schmuller, 2002).

Diagrama de casos de uso

Un CU describe una función del software, desde el punto de vista del usuario.

Esta técnica permite visualizar errores y aciertos a la hora de determinar los requisitos del sistema. Las personas y entes externos que intervienen con el software se representan con una figura de monigote denominado actor, ubicado fuera de la barra de límite del sistema.

Dentro de un diagrama de CU, cuando un actor o un CU posee las mismas cualidades que otro, se usa una relación de generalización hacia el más amplio. Un CU puede incluir la funcionalidad de otro como parte de su procesamiento normal, ayudando así a reducir la duplicación de funcionalidad y puede extender el comportamiento de otro, típicamente cuando ocurren situaciones excepcionales. Su notación se muestra en la Ilustración 3.

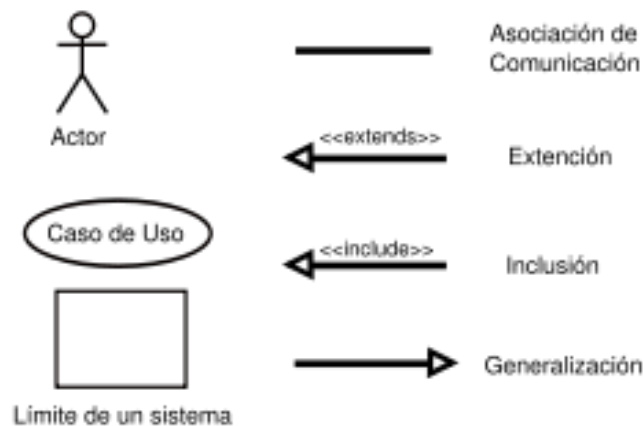


Ilustración 3: Notación del diagrama de caso de uso

Diagrama de clases

Este diagrama se usa para especificar el patrón de lo que producirán los objetos en tiempo de ejecución y está formado por varios rectángulos, que hacen referencia a las clases, conectados por líneas que muestran las relaciones entre ellas. Representa información estática. Su notación se muestra en la

Ilustración 4.

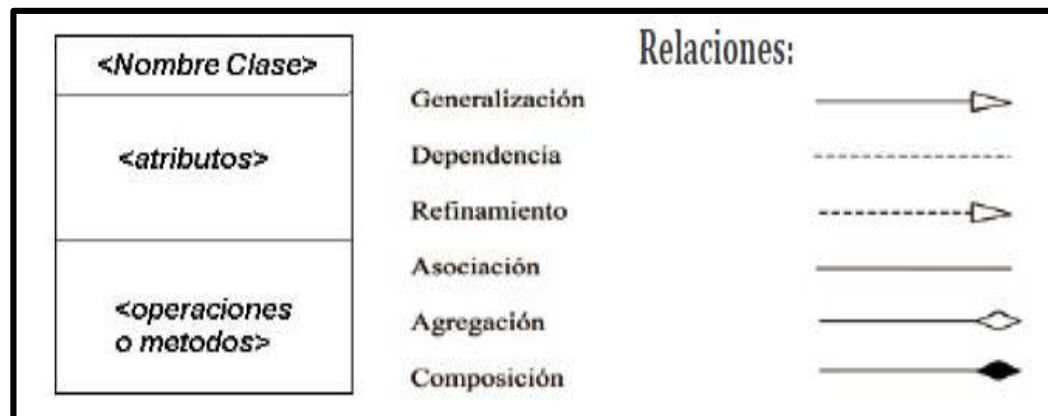


Ilustración 4: Notación del diagrama de clases

1.8.3.5. ENTORNO DE DESARROLLO

Los Entornos Integrados de Desarrollo, abreviado IDE por sus siglas en inglés (Integrated Development Environment) son aplicaciones de software que ayudan al programador en el desarrollo de software.

Generalmente todos los IDEs contienen al menos un editor de código fuente, herramientas de compilación y un debugger (depurador). Actualmente, los IDEs cuentan con más herramientas, como por ejemplo función de autocompletar el código mientras se va escribiendo. Los entornos de desarrollo han sido pensados para aumentar la productividad del programador. Esto se puede lograr al integrar en un sólo programa de desarrollo las herramientas más comunes que usa el programador. Esto reduce el tiempo de configuración antes del desarrollo de un proyecto. Algunos IDEs son específicos para un lenguaje en particular, aunque la mayoría soportan múltiples lenguajes. Los IDEs facilitan generar interfaces gráficas de usuario. Muchos IDEs permiten al programador crear interfaces gráficas arrastrando elementos de la interfaz y definiendo sus características visualmente, sin la necesidad de escribir código. Debido a que este proyecto se hará usando Java, hay que decidir qué entorno de desarrollo para utilizar Java. Para este proyecto se usará Netbeans, pues facilita en gran medida la creación de interfaces gráficas, además de tener otras características que se mencionarán a continuación. Netbeans es un entorno de desarrollo escrito en Java que permite programar en múltiples lenguajes de programación. Cuenta con diferentes Plugins que se pueden descargar dependiendo de las herramientas que necesitemos para nuestro desarrollo. Dichos Plugins son desarrollados por la comunidad de Netbeans, e incluyen varias categorías, por ejemplo:

- Soporte para diversos lenguajes de programación
- Herramientas de edición
- Soporte para bases de datos
- Depuración
- Herramientas de control de versiones
- Desarrollo web
- Utilidades varias

Estos Plugins facilitan al programador varias tareas. En el caso de este proyecto, se utilizó un Plugin para Mercurial, sistema de control de versiones, de modo que el control de versiones se pueda llevar a cabo desde el mismo entorno de desarrollo. Sobre este sistema de control de versiones se hablará en la sección.

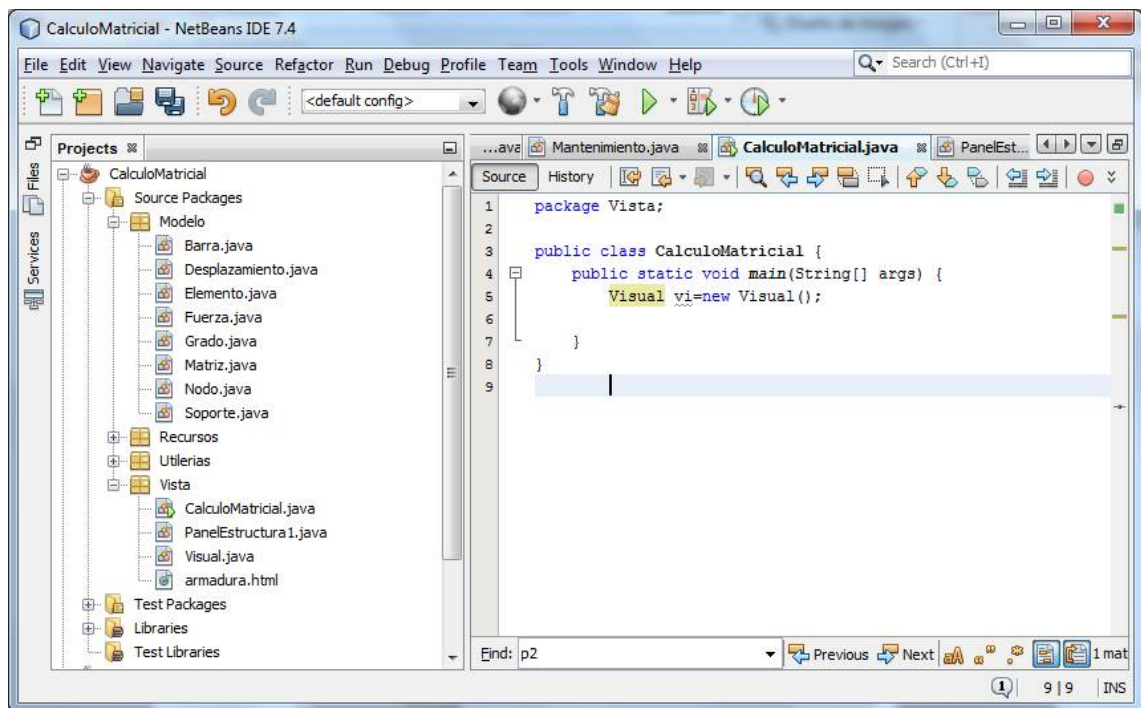


Ilustración 5: Ventana de Netbeans

En la ilustración 5 se observa una interfaz gráfica creada en Netbeans. Como se puede observar, del lado derecho se encuentran todos los elementos disponibles para implementar una interfaz gráfica. Basta con arrastrar estos elementos hacia la pantalla del centro para ir acomodándolos dentro de la interfaz. Hay que resaltar que como se mencionó, el IDE es especialmente útil para manejar elementos de la interfaz gráfica, pues se pueden agregar y editar de manera visual. Esto nos ahorra tiempo en el desarrollo de la Vista, un elemento de la arquitectura MVC, de la que se hablará en la siguiente sección.

1.8.3.6. ARQUITECTURA MODELO VISTA CONTROLADOR

Esta arquitectura está pensada para el desarrollo de interfaces gráficas de usuario. Con esta arquitectura, se puede llevar un mejor control de los elementos que componen al sistema, de modo que sea más fácil el desarrollo y mantenimiento del mismo. Este tipo de arquitectura o patrón de diseño está muy ligado a la programación orientada a objetos, pues se basa en la abstracción y también en la reutilización de código. El MVC tiene tres componentes que interactúan entre sí: el modelo, la vista y el controlador (ver ilustración 6).

Modelo. -Representa los datos de la aplicación y las reglas, la lógica y las funciones. El modelo notifica a la vista y al controlador cuando hay un cambio en el estado de la aplicación. que gobiernan el acceso a y la actualización de los datos.

Vista. - Especifica cómo deben ser presentados los datos del modelo. Si los datos del modelo cambian, la vista debe actualizar la presentación de los datos en el momento. Para esto se puede usar un push model, donde el modelo manda los cambios en todo momento, o un pull model, donde la vista es responsable de llamar al modelo cuando necesite actualizar a los datos más recientes.

Controlador. - Se encarga de interpretar las acciones que el modelo debe efectuar, cuando el usuario interactúa con la Vista. Estas interacciones son los clics del mouse, una selección de menú o un GET en un mensaje de HTTP, por ejemplo, dependiendo de la aplicación de la que se trate.

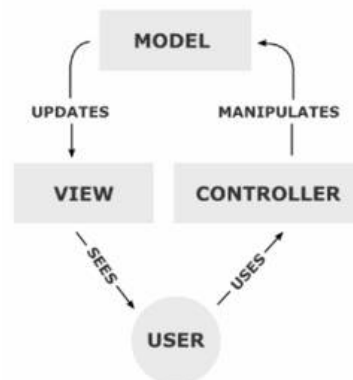


Ilustración 6: Modelo Vista Controlador

Un ejemplo de la aplicación del MVC en Java está en sus mismas bibliotecas de gráficos. Tal es el caso de Swing, una biblioteca gráfica de Java, en la que los componentes se comportan de acuerdo al MVC como se muestra a continuación:

Modelo. - Cada componente de Swing tiene modelos asociados, que pueden ser usados por el programador tal cual están, o implementar sus propios modelos usando las interfaces de Java. Por ejemplo, para un botón en Java, el modelo definiría los eventos (presionar, liberar, etcétera) y las propiedades del botón (color, texto, etcétera).

Vista. - En un componente de Swing, la vista sería lo que ve el usuario en pantalla y está implementado en su mayoría en los métodos `paint()` de los componentes.

Controlador. - El controlador se encargará de detectar los eventos que surjan tras la interacción del usuario con la vista. Esto eventos son detectados al implementar `Listeners` de Java para los eventos específicos.

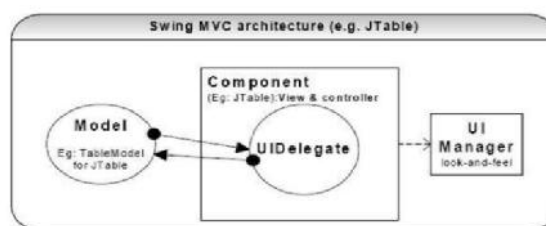


Ilustración 7

Como se puede observar, en el caso de Swing (ilustración 7) la vista y el controlador están muy ligados, por lo que se encuentran incluso en la misma clase. Debido a que se trata de un patrón de diseño abstracto, es posible implementarlo de diferentes maneras. En Java, una de las implementaciones que se puede hacer es creando directamente las clases para el Modelo, la Vista y el Controlador. Esto aplicaría para un proyecto pequeño, en el que no hay muchos elementos que manejar y en el que fueran suficientes estas tres clases para todo el sistema. Pero para un sistema más grande, este enfoque no sería conveniente.

La alternativa es usar paquetes para organizar las clases dentro de las categorías de Modelo, Vista y Controlador. Éste es el enfoque que se usará para este proyecto, pues como se verá más adelante, el proyecto tiene un gran número de clases. Si no se llevara este orden, sería muy difícil manejar tantas clases. Sin embargo, cabe mencionar que el paquete para la Vista contendrá el código tanto para la Vista como para el Controlador, pues como vimos anteriormente, ambos elementos están muy ligados.

1.8.3.7. CONTROL DE VERSIONES

Durante todas las etapas del proceso de desarrollo es importante documentar los cambios que se van haciendo conforme se va avanzando en el proyecto. De esta manera se puede cuantificar los avances conforme transcurre el tiempo y saber si se tendrá el software en el tiempo estimado. Esto se puede realizar simplemente llevando una bitácora de los cambios que se van haciendo, aunque hay herramientas que

permiten hacer esto y agregan otras características interesantes. A este tipo de herramientas se les llama sistemas de control de versiones o System Version Control (SVC) en inglés. Con estos sistemas podemos ir documentando el avance del proyecto, pero además nos permite hacer un regreso a una versión anterior en caso de que se requiera. Entre los SVCs más populares tenemos:

CVS (Control Version System): muy popular para desarrollo de software libre. Usa la licencia GPL.

Subversion: esta herramienta es de código abierto y muy popular para el desarrollo de software de código abierto

Team Foundation Version Control: es una herramienta que viene integrada en la suite de desarrollo de Visual Studio de Microsoft.

Bazaar: esta herramienta es patrocinada por Canonical Ltd, la empresa creadora de Ubuntu.

Git: este SVC fue diseñado por Linus Torvalds, el creador del núcleo de Linux, pensado para proyectos de código abierto en donde el número de archivos es muy extenso y también para el desarrollo colaborativo.

Mercurial: originalmente creado para Linux, pero ahora funciona en MacOS y Windows también. Está pensado para hacer un desarrollo distribuido, que generalmente se da en proyectos de software libre o de código abierto.

1. Mercurial

Para este proyecto se usó Mercurial, pues este proyecto podría ser un buen candidato para ser de código abierto. Un proyecto de código abierto es aquel en el que la licencia permite al usuario modificar el código y compilarlo para crear su propia versión de él. Los proyectos de código abierto se benefician de la interacción de múltiples programadores que pueden aportar ideas y colaborar a través de Internet. Una ventaja de Mercurial es que está integrado con Netbeans, lo que lo hace fácil de usar si ya se conoce la plataforma de Netbeans. Para llevar el control de versiones simplemente hay que ir haciendo commits, conforme se va avanzando con el proyecto. Los commits indican al Mercurial que se deben considerar los cambios que se hicieron al programa como definitivos, por lo que el commit creará una versión del programa. Para hacer un commit desde Netbeans se da clic derecho sobre el nombre del proyecto y en Mercurial-

>Commit (Ilustración 8).

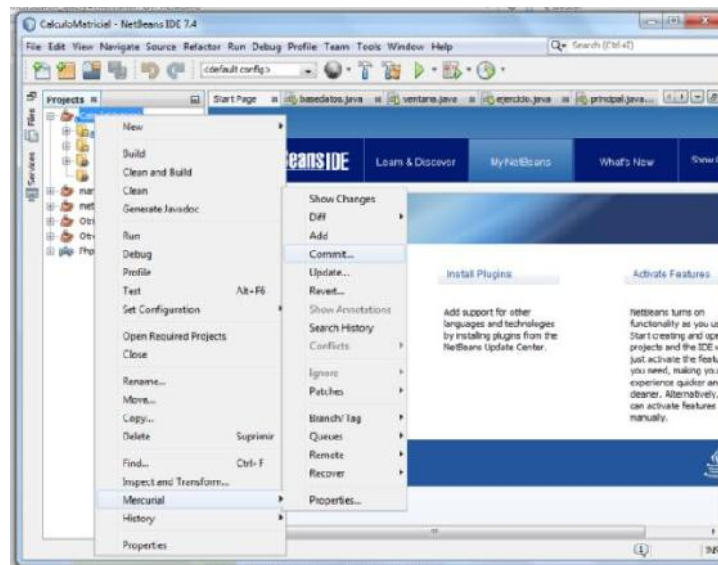


Ilustración 8: Acceder a Commit desde Netbeans

Al hacer un commit se debe agregar una descripción de los cambios que se hicieron (ver ilustración 9) para que quede registrado en la bitácora. Esta descripción debe ser breve y concisa para que en caso de que el programador tenga que consultar alguna de las versiones pueda ver rápidamente que cambios se realizaron en ella.

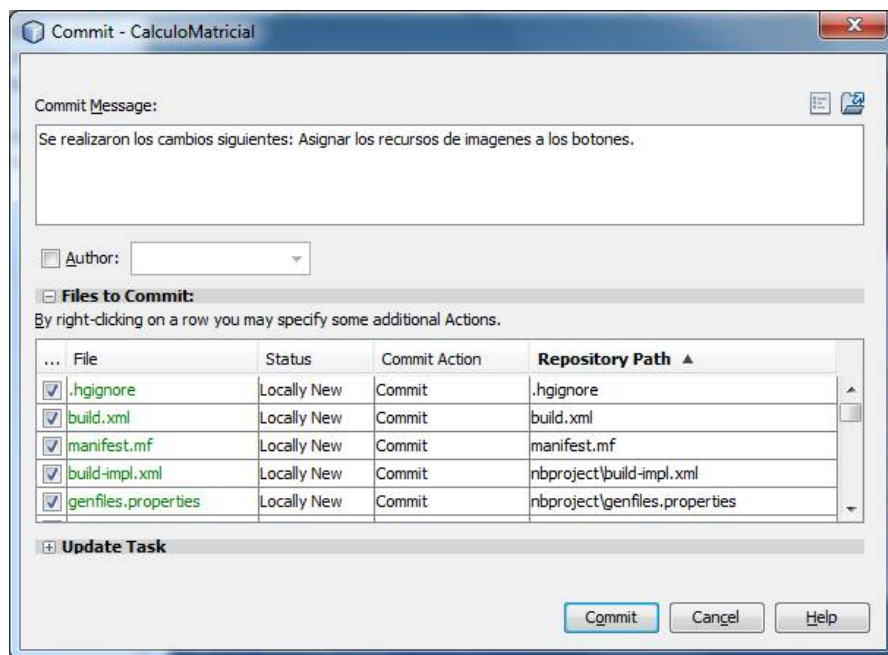


Ilustración 9: Commit para el SAEMEF

Al consultar el histórico de las versiones (ilustración 10), se puede ver la descripción que se puso y las diferencias entre los archivos de esa versión y la versión anterior. Si se usa un repositorio, todos estos datos se podrán consultar también desde ahí.

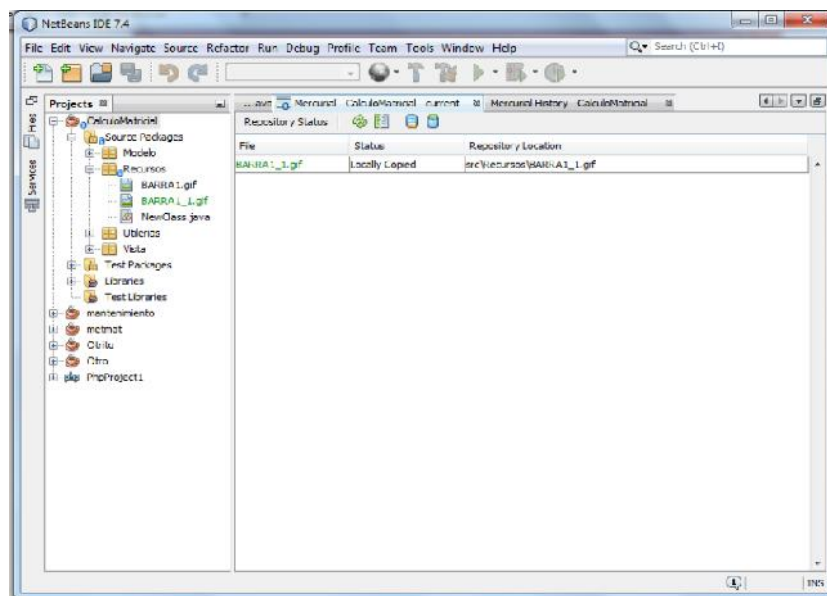


Ilustración 10: Histórico de las versiones.

1.8.3.8. REPOSITORIO

Además del control de versiones, es importante tener un método de respaldo para el proyecto. Esto a veces no se tiene en cuenta y por algún motivo externo se puede perder todo el avance de un proyecto. Hay veces que por alguna situación ajena a nosotros el equipo en donde se está trabajando el proyecto se descompone. Ya sea por una falla eléctrica, deterioro de alguno de los componentes o por un descuido (derramar el café, por ejemplo) el equipo puede quedar dañado y la información contenida perdida para siempre. Una manera de hacer respaldos es simplemente guardando el proyecto en CDs o memorias externas, de modo que, si hay algún problema, podamos recuperar, aunque sea una versión antigua del proyecto. Para asegurar que en caso de falla la pérdida sea mínima, es importante hacer respaldos constantes (diarios, semanales o mensuales).

Otra manera un poco más profesional de hacer respaldos, es a través del uso de repositorios. Un repositorio es una estructura de datos almacenada en un servidor que contiene un conjunto de archivos y directorios, y una bitácora de cambios en el repositorio. Generalmente en las empresas, donde el código es

cerrado y es un bien que debe resguardarse, los repositorios se crean en servidores internos que no pueden ser accedidos desde fuera. Sin embargo, para proyectos de software libre, código abierto y hasta proyectos escolares, hay muchos sitios que ofrecen el servicio de repositorio desde Internet. De esta manera, lo único que se necesita para subir una versión del proyecto al repositorio es una conexión a Internet.

Un repositorio como ejemplo es el servicio de Bitbucket. Bitbucket con un servicio de alojamiento de código fuente para proyectos que usan Mercurial o Git como sus sistemas de control de versiones. En sus cuentas gratuitas, se permite la creación de los repositorios que se deseen y permite elegir si el repositorio se quiere hacer privado o público, de modo que otras personas puedan descargar el proyecto y hacer modificaciones, pero sin la posibilidad de subir esas modificaciones. Para que otras personas puedan subir cambios al repositorio se necesita tener una cuenta de pago, con lo que el repositorio podría ser accedido por un número determinado de usuarios.

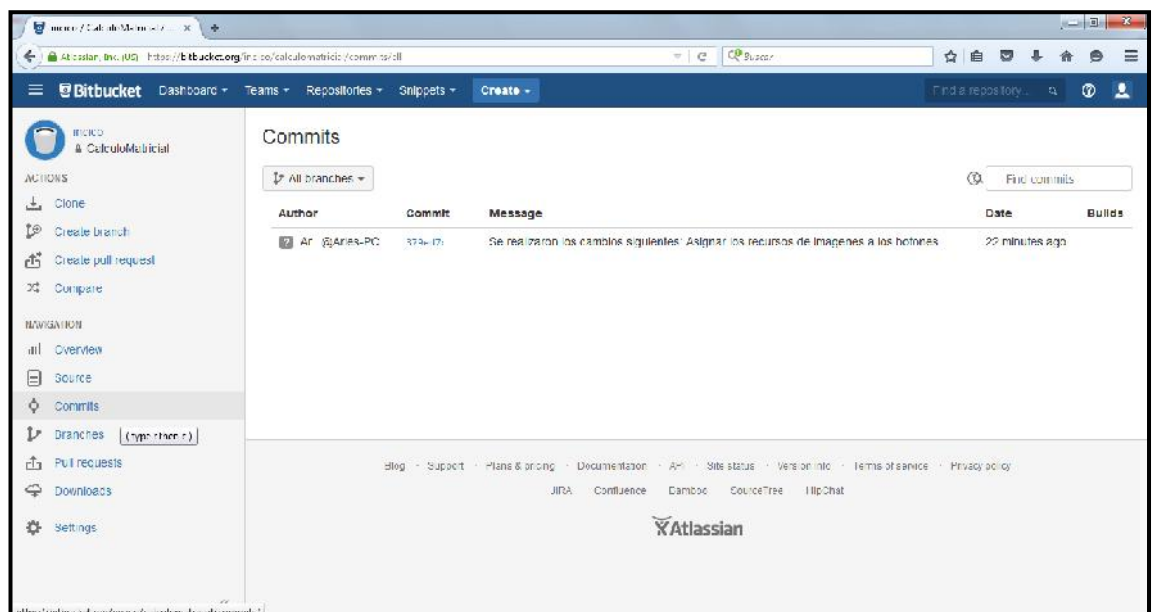


Ilustración 11: Repositorio

En la pantalla anterior (ilustración 11) se muestra el repositorio que se creó para el proyecto. Como se puede observar, aparece que el usuario Aries realizó una commits. Estos commits son los que se hacen desde mercurial. Para subir el proyecto al repositorio se tiene que hacer un push desde Netbeans. Para esto se da clic derecho en el nombre del proyecto y luego en Mercurial->Remote->Push. El Plugin de Mercurial debe

estar instalado dentro de Netbeans para que aparezca esta opción. Netbeans mostrará la siguiente ventana (ilustración 12):

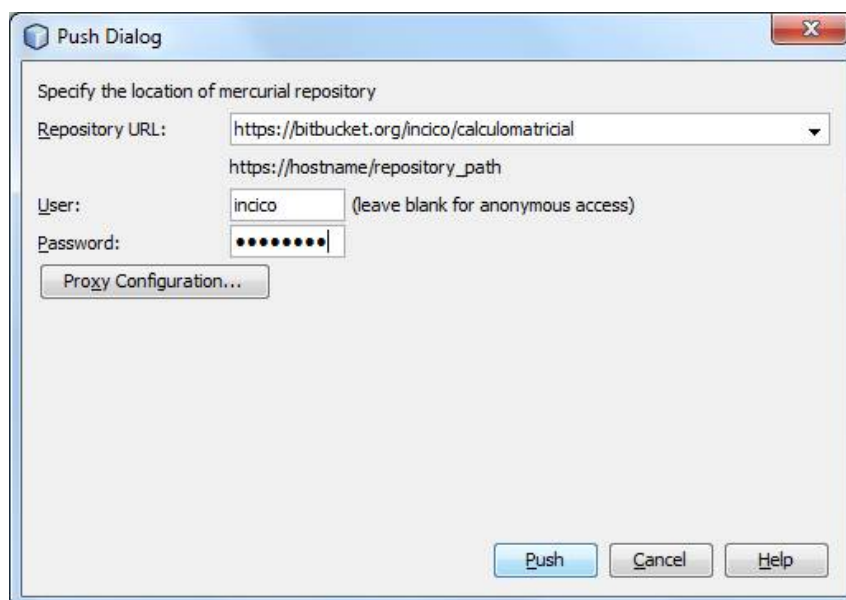


Ilustración 12

En dicha ventana se debe escribir la dirección del repositorio del proyecto, así como el usuario y contraseña de un usuario del repositorio. Al presionar el botón Push el proyecto se subirá al repositorio. De esta manera tendremos respaldado el proyecto. Para descargar el proyecto del repositorio simplemente se hace un pull desde el menú Team->Remote->Pull. Se pedirán los mismos datos que para el push.

1.8.3.9. MATERIAL EDUCATIVO COMPUTARIZADO MEC

En el medio digital existen varios tipos de programas de aplicación que se ejecutan en un computador, de los cuales solo algunos corresponden a la categoría de Materiales Educativos Computarizados (MECs). Dentro de los programas educativos es importante señalar los tipos de programas que los usuarios educativos pueden encontrar.

TIPOS DE MEC

Según el enfoque educacional que da soporte a la misma y de la función como intentan apoyar los existe de varios tipos.

Thomas Dwyer, define una gran clasificación de los MECs considerando el enfoque educativo de los cuales predominan los: algorítmicos y el heurísticos.

Un material de tipo algorítmico es aquel en que predomina el aprendizaje vía transmisión de conocimiento, desde quien sabe hacia quien lo desea aprender y donde el diseñador se encarga de encapsular secuencias bien diseñadas de actividades de aprendizaje que conducen al aprendiz desde donde está hasta donde desea llegar. El rol del alumno es asimilar el máximo de lo que se le transmite.

Un material de tipo heurístico es aquel en el que predomina el aprendizaje experimental y por descubrimiento, donde el diseñador crea ambientes ricos en situaciones que el alumno debe explorar conjeturalmente. El alumno debe llegar al conocimiento a partir de la experiencia, creando sus propios modelos de pensamiento, sus propias interpretaciones del mundo, las cuales puede someter a prueba con el MEC.

Otra forma de clasificarlos es por las funciones mismas que asumen, claro está, ubicables dentro de la taxonomía antes presentada. A nivel algorítmico podemos distinguir los sistemas tutoriales y los sistemas de ejercitación y práctica. Dentro de la categoría heurística se distinguen los simuladores, los juegos educativos, los lenguajes sintónicos y algunos sistemas expertos.

En cualquiera de las dos grandes categorías pueden ubicarse los sistemas inteligentes de aprendizaje apoyado con computador, dependiendo del rol que esté asumiendo el material.

1.8.3.10. METODOLOGÍA PARA EL DESARROLLO DEL MEC.

Las metodologías para el desarrollo de MEC's a través de procesos sistemáticos hacen un particular énfasis en los siguientes aspectos: la solidez del análisis como punto de partida; el dominio de teorías sobre el aprendizaje y la comunicación humana, como base teórica para el diseño del ambiente educativo

computarizado; la evaluación permanente a lo largo de todas las etapas del proceso, como medio de perfeccionamiento continuo del material; la documentación adecuada y suficiente de lo que se realiza en cada una de las etapas, como base para el mantenimiento que se dará al material a lo largo de su vida útil.

ANÁLISIS DE NECESIDADES EDUCATIVAS

Todo MEC debe jugar un papel importante donde se utilice; su incorporación en el proceso de enseñanza no se debe a que “está bonito” o “se encontró”, sino que su utilización refleja máximos beneficios a la comunidad educativa.

Para realizar el análisis de las necesidades educativas se toman en cuenta:

- Consulta a fuentes de información apropiadas e identificación del problema.
- Análisis de posibles causas de los problemas detectados.
- Análisis de alternativas de solución.
- Establecimiento del papel del computador.
- Selección o planeación del desarrollo de MEC's.

DISEÑO DEL MEC

El diseño de un MEC se fundamenta en los resultados arrojados en la etapa de análisis.

Para llevarse a cabo el Diseño del MEC, se toman en cuenta los siguientes aspectos:

- **Entorno para el diseño del MEC.** A partir de los resultados del análisis, es conveniente señalar las características del MEC a diseñar.

Según Álvaro Galvis (1992), en el entorno del Diseño se busca dar solución a las siguientes interrogantes asociadas a dicha especificación:

“¿A quiénes se dirige el MEC?, ¿Qué características tienen sus destinatarios?

¿Qué área de contenido y unidad de instrucción se beneficia con el estudio del MEC?

¿Qué problemas se pretende resolver con el MEC?

¿Bajo qué condiciones se espera que los destinatarios usen el MEC?

¿Para un equipo con qué características físicas y lógicas conviene desarrollar el MEC?”. (p.71)

- **Diseño Educativo del MEC.** En el diseño educativo se trata el alcance, contenido y tratamiento que debe apoyar el MEC.

Según Alvaro Galvis, con las siguientes interrogantes se interesa resolver el diseño educativo del MEC:

“¿Qué aprender con apoyo del MEC?

¿En qué ambiente o micro-mundo aprenderlo?

¿Cómo motivar y mantener motivados a los usuarios del MEC?

¿Cómo saber que el aprendizaje se está logrando?”. (p.72)

- **Diseño de Comunicación.** Conocida como a interacción entre usuarios y programas, la interfaz es otro punto a diseñar de los MEC's.

Con las siguientes interrogantes Alvaro Galvis hace referencia a las características que debe poseer el Diseño de Comunicación:

“¿Qué dispositivos de entrada y salida conviene poner a disposición del usuario para que se intercomunique con el MEC?

¿Qué zonas de comunicación entre usuarios y programa conviene poner a disposición en y alrededor del micro mundo seleccionado?

¿Qué características debe tener cada una de las zonas de comunicación?

¿Cómo verificar que la interfaz satisface los requisitos mínimos deseables?” (p.72)

Diseño Computacional. Con base en las necesidades previamente identificadas, se establece que funciones se desea que cumpla el MEC en apoyo a sus usuarios, y/o estudiantes. El MEC puede brindar al alumno la posibilidad de controlar el ritmo, la cantidad de contenido y ejercicios, de abandonar o reiniciar.

Además, un MEC ofrece al profesor la posibilidad de editar ejercicios, contenidos, manejar el registro de estudiantes y el rendimiento que demuestra con el MEC, etc.

Los componentes principales del diseño computaciones los refleja Alvaro Galvis de la siguiente manera:

“¿Qué funciones se requiere que cumpla el MEC para cada uno de los tipos de usuarios?

Para el módulo del profesor y para el del estudiante, ¿Qué estructura lógica comandará las acciones?

¿Qué estructuras de datos, se necesitan para que funcione el MEC?” (p. 73)

- **Preparación y Revisión de un Prototipo del MEC.** Para finalizar el diseño se analiza el prototipo elaborado verificándose si tiene sentido frente a la necesidad de la población a la que se dirige el MEC.

Los prototipos consisten en la elaboración de bocetos en papel de cada uno de los ambientes (pantallas) que formarán la estructura lógica y las acciones de los eventos que contendrá el MEC.

DESARROLLO DEL MEC.

Una vez obtenido el diseño debidamente documentado, se lleva a cabo su implementación (desarrollo), cumpliendo las metas en términos de tiempo y calidad del MEC.

- **Estrategias para el desarrollo de MEC's.**

Las estrategias usadas para el desarrollo del MEC, dependen de los recursos humanos y computacionales con los que se cuenta; es decir:

- Si se cuenta con un grupo interdisciplinario (especialistas en contenido, metodología e informática), el desarrollo recaerá sobre el especialista de informática, pero contará con el resto de especialistas para el buen desarrollo del MEC. Así, a medida que el especialista en informática va desarrollando los módulos, los demás miembros del equipo de especialistas lo van revisando para ajustar lo que se requiera.
- Cuando no hay un especialista en informática o quien sepa programar en un lenguaje de propósito general, cabe considerar 2 alternativas:
 - ✓ Contratar la programación del diseño con un especialista de informática externo.
 - ✓ Intentar que los miembros del equipo se animen y aprendan a usar un lenguaje o un sistema autor, de modo que ellos mismos manipulen y trabajen en el desarrollo del MEC.

- **Desarrollo y Documentación del MEC**

Indistintamente de la estrategia empleada para el desarrollo del MEC es fundamental programar en forma estructural y legible, así como documentar el trabajo.

La documentación esperada es:

- Dentro del programa, dar nombre significativo a los procedimientos, codificar variables y constantes según la estructura definida.
- La documentación en un manual de usuarios.
- Un manual para mantenimiento.

- **Revisión del MEC mediante juicio de expertos**

Es importante verificar con base en el diseño, si lo previsto se ha llevado a la práctica o si los ajustes que se introdujeron en el desarrollo efectivamente mejoran la calidad del diseño.

Es importante señalar que la revisión por expertos no implica, necesariamente, que el MEC va a funcionar apropiadamente y producir los resultados esperados al ser usado por los destinatarios.

- **Revisión uno a uno con usuarios representativos**

Los usuarios son los únicos que pueden decir si un MEC está bien logrado o no.

Por este motivo conviene que, como una de las etapas finales del desarrollo, se realice la revisión del MEC con unos pocos usuarios representativos. Dicha revisión se hace al ritmo del usuario, estando acompañado por alguien del grupo desarrollador.

Como es evidente, ésta revisión es para detectar problemas de interfaz o de otra índole y corregirlos, no para asegurar que el MEC es efectivo.

PRUEBA PILOTO DEL MEC.

En esta prueba se pretende ayudar a la depuración del MEC a partir de su utilización por una muestra representativa de los tipos de destinatarios para los que se hizo y la consiguiente evaluación formativa.

- **Preparación de la prueba piloto**

Esta incluye la recolección de la muestra, el diseño y la prueba de los instrumentos de recolección de información y el entrenamiento de quienes van a administrar la prueba del material.

- **Desarrollo de la prueba piloto**

Se seleccionan los aprendices, dejándolos interactuar con el material a su gusto, disponiendo del tiempo que requieran hasta que crean dominar los objetivos.

Cuando los usuarios crean haber aprovechado al máximo el MEC, se le aplica una prueba final equivalente a la prueba previa, para establecer cuánto aprendieron y poder determinar en qué está fallando cada quien.

Estas sesiones de trabajo, permiten establecer el tiempo mínimo, máximo y promedio que se requiere para estudiar el MEC y sacar provecho de él.

- **Toma de decisiones acerca del MEC**

Dependiendo de los resultados de las pruebas se pueden tomar decisiones como las siguientes:

1. Desechar el MEC.
2. Ajustar detalles en el MEC.
3. Hacer ajustes mayores en el MEC.

PRUEBA DE CAMPO DEL MEC.

La prueba de campo de un MEC es mucho más que usarlo con toda la población objeto. Dentro del ciclo de desarrollo de un MEC hay que buscar la oportunidad de comprobar, en la vida real, que aquellos que a nivel experimental parecían tener sentido, lo siguen teniendo.

1.8.3.11. ANÁLISIS ESTRUCTURAL

Análisis estructural se refiere al uso de las ecuaciones de la resistencia de materiales para encontrar los esfuerzos internos que actúan sobre una estructura resistente, como edificaciones o esqueletos resistentes de maquinaria. El análisis estructural consiste en la determinación de los efectos originados por las acciones sobre la totalidad o parte de la estructura, con objeto de efectuar comprobaciones en los Estados Límite Últimos y de Servicio.

1.8.3.12. MODELOS ESTRUCTURALES

Para la realización del análisis, se idealizan tanto la geometría de la estructura como las acciones y las condiciones de apoyo mediante un modelo matemático adecuado. El modelo elegido deberá ser capaz siempre de reproducir el comportamiento estructural dominante. Para el análisis, los elementos estructurales se clasifican en unidimensionales, cuando una de sus dimensiones es mucho mayor que las restantes, bidimensionales, cuando una de sus dimensiones es pequeña comparada con las otras dos, y tridimensionales cuando ninguna de sus dimensiones resulta sensiblemente mayor que las otras.

COLUMNA

Elemento estructural que se usa principalmente para resistir carga axial de compresión y que tiene una altura de por lo menos 3 veces su dimensión lateral menor.

VIGA

Elemento estructural que trabaja fundamentalmente a flexión.

1.8.3.13. MÉTODO DE LOS ELEMENTOS FINITOS

El método de los elementos finitos (MEF en castellano o FEM en inglés) es un método de cálculo utilizado en diversos problemas de ingeniería, que se basa en considerar al cuerpo o estructura dividido en elementos discretos, con determinadas condiciones de vínculo entre sí, generándose un sistema de ecuaciones que se resuelve numéricamente y proporciona el estado de tensiones y deformaciones. También

se utiliza en matemáticas como método nodal aproximado para resolver ecuaciones diferenciales en forma numérica.

Es un procedimiento numérico aplicable a un gran número de problemas con condiciones de borde impuestas (en las estructuras las condiciones de borde serían: restricciones y cargas externas). Varios de estos problemas no tienen solución analítica o es muy difícil obtenerla, por lo que se convierte en la única alternativa de resolución. Con este método se pueden resolver sistemas los cuales no son fáciles de resolver mediante modelos matemáticos simples. Existen dos tipos de caminos para su formulación, basándose en el principio de los trabajos virtuales, es decir, formulaciones variacionales, o mediante el método de Galerkin, Método directo o bien con Raleigh Ritz.

Si bien fue originalmente desarrollado para el análisis de estructuras, con este método se pueden representar entre otros, los siguientes fenómenos físicos:

- Fenómenos termodinámicos: distribución de temperaturas en un sólido.
- Simulación de efectos dinámicos: Choque de dos cuerpos.
- Geo mecánica: Comportamiento de la corteza terrestre.

ELEMENTOS GENERALES A TRAVÉS DE FEM

Los tipos de elementos utilizados generalmente en la resolución a través de FEM son:

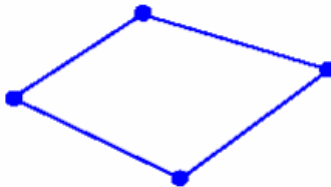
- Elementos Lineales (1-D)



Estos pueden ser:

- Resorte
- Barras
- Vigas
- Caños

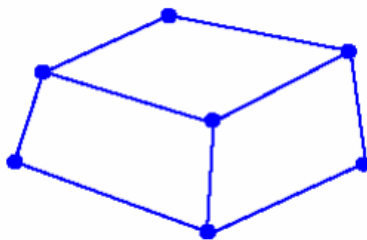
- Elementos Planos (2-D)



Estos pueden ser:

- Membranas
- Placas

- Elementos Sólidos (3-D)

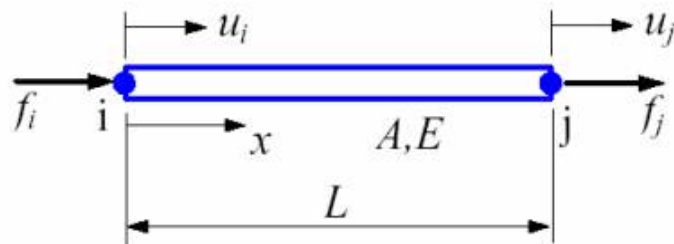


Es importante destacar que se puede utilizar combinaciones de estos elementos actuando en conjunto.

MÉTODO DIRECTO

A.-Elemento Barra en Una Dimensión

Consideremos una barra de sección constante:



El sistema se compone de:

- Dos Nodos: i, j
- Modulo de Elasticidad E
- Área de la Sección Transversal A
- Longitud del Elemento L

El mismo está sometido:

- Fuerzas en los Nodos: f_i, f_j

El elemento tiene dos grados de libertad, en el sentido longitudinal del elemento, cualquier desplazamiento de los nodos en el sentido normal al elemento no generara esfuerzos internos:

- Dos desplazamientos: u_i, u_j

Sabiendo que la rigidez a tracción / compresión de una barra es:

$$\frac{EA}{L}$$

Y haciendo una analogía con el elemento resorte, tenemos que:

$$k = \frac{EA}{L}$$

Por lo tanto,

$$\mathbf{k} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} = \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix}$$

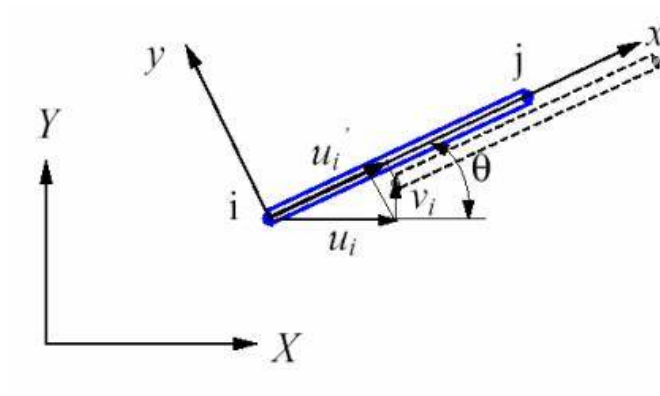
O bien,

$$\mathbf{k} = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Por lo tanto, la ecuación de equilibrio del Elemento será:

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \begin{Bmatrix} f_i \\ f_j \end{Bmatrix}$$

B.-Elemento Barra en Dos Dimensiones



Local	Global
x,y	X,Y
u_i', v_i'	u_i, v_i
1 grado de Libertad	2 grados de Libertad

Nota: El desplazamiento lateral v_i' no contribuye a la deformación de la barra.

La idea es trabajar con las coordenadas globales, por lo tanto se deben hacer las siguientes transformaciones:

$$u_i' = u_i \cos \theta + v_i \sin \theta = \begin{bmatrix} l & m \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}$$

$$v_i' = -u_i \sin \theta + v_i \cos \theta = \begin{bmatrix} -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}$$

Donde:

$$l = \cos \theta, \quad m = \sin \theta$$

Escrito en forma matricial:

$$\begin{Bmatrix} u_i' \\ v_i' \end{Bmatrix} = \begin{bmatrix} l & m \\ -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}$$

O bien,

$$\mathbf{u}_i' = \tilde{\mathbf{T}} \mathbf{u}_i$$

Donde la matriz transformación:

$$\tilde{\mathbf{T}} = \begin{bmatrix} l & m \\ -m & l \end{bmatrix}$$

Donde la relación con la matriz ortogonal:

$$\tilde{\mathbf{T}}^{-1} = \tilde{\mathbf{T}}^T$$

A modo de ejemplo, se puede decir que para un sistema de elemento barra con dos nodos tenemos que:

$$\begin{Bmatrix} u_i' \\ v_i' \\ u_j' \\ v_j' \end{Bmatrix} = \begin{bmatrix} l & m & 0 & 0 \\ -m & l & 0 & 0 \\ 0 & 0 & l & m \\ 0 & 0 & -m & l \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

O bien;

$$\mathbf{u}' = \mathbf{T} \mathbf{u}$$

Con

$$\mathbf{T} = \begin{bmatrix} \tilde{\mathbf{T}} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{T}} \end{bmatrix}$$

Las fuerzas nodales son transformadas de la misma manera:

$$\mathbf{f}' = \mathbf{T} \mathbf{f}$$

Para obtener la matriz rigidez en dos dimensiones:

En el sistema local de coordenadas tenemos que:

$$\frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \begin{Bmatrix} f_i \\ f_j \end{Bmatrix}$$

Esto se puede escribir en su totalidad como:

$$\frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} = \begin{Bmatrix} f_i \\ 0 \\ f_j \\ 0 \end{Bmatrix}$$

O bien,

$$\mathbf{k}' \mathbf{u}' = \mathbf{f}'$$

Utilizando las transformaciones

$$\mathbf{u}' = \mathbf{T} \mathbf{u}$$

$$\mathbf{f}' = \mathbf{T} \mathbf{f}$$

Obtenemos:

$$\mathbf{k}' \mathbf{T} \mathbf{u} = \mathbf{T} \mathbf{f}$$

Multiplicando ambos lados por \mathbf{T}^T y como $\mathbf{T}^T \mathbf{T} = \mathbf{I}$, obtenemos:

$$\mathbf{T}^T \mathbf{k}' \mathbf{T} \mathbf{u} = \mathbf{f}$$

La matriz rigidez en el sistema global quedara de la siguiente manera:

$$\mathbf{k} = \mathbf{T}^T \mathbf{k}' \mathbf{T}$$

La cual es una matriz simétrica de 4X4. Escrita de manera explicita tenemos que:

$$\mathbf{k} = \frac{EA}{L} \begin{bmatrix} u_i & v_i & u_j & v_j \\ l^2 & lm & -l^2 & -lm \\ lm & m^2 & -lm & -m^2 \\ -l^2 & -lm & l^2 & lm \\ -lm & -m^2 & lm & m^2 \end{bmatrix}$$

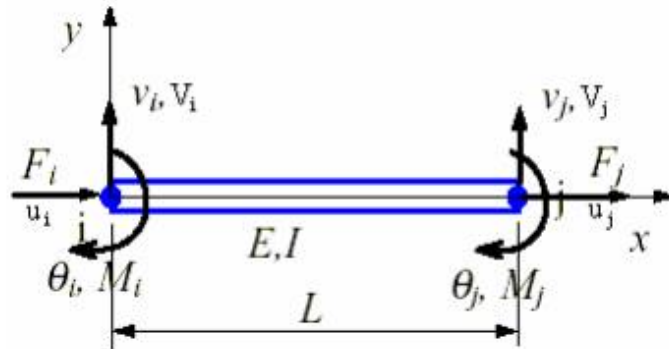
Donde los cosenos directores l y m son:

$$l = \cos \theta = \frac{X_j - X_i}{L}, \quad m = \sin \theta = \frac{Y_j - Y_i}{L}$$

C.-Elemento Viga

Se considera una viga en el plano. Esta toma esfuerzos de Corte, Axiles y Momentos, todas consideradas en el plano. Cada Nodo posee tres Grados de Libertad (u, v, q).

Un elemento que toma estas cargas, tiene asociado para el cálculo a E, J, I y A .



El sistema se compone de:

- Dos Nodos: i, j
- Modulo de Elasticidad E
- Área de la Sección Transversal A
- Longitud del Elemento L
- Momento de Inercia I

El mismo está sometido:

- Fuerzas en los Nodos: F_i, F_j, V_i, V_j
- Momento en los Nodos: M_i, M_j

Habrán tres grados de libertad por cada nodo

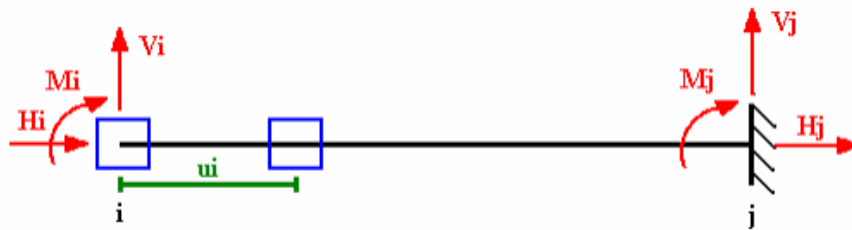
- Cuatro desplazamientos: u_i, u_j, v_i, v_j
- Dos Giros: θ_i, θ_j

Para crear la Matriz Rigidez se suponen casos con desplazamientos unitarios, que luego mediante Superposición se ensamblan y dan forma a dicha matriz. Se adoptan giros en sentido horario y desplazamientos positivos.

Se supone

$$u_i = 1$$

$$u_j, v_i, v_j, \theta_i, \theta_j = 0$$



Aplicando la Ley de Hooke, tal como se hace con elemento barra, tenemos que:

$$H_i = \frac{EA}{l} \cdot u_i$$

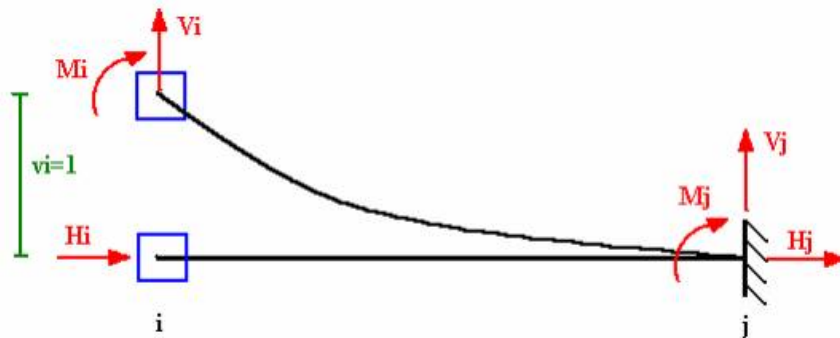
Por lo tanto,

$$H_i = \frac{EA}{l}$$

Realizando un equilibrio de fuerzas,

$$H_j = -\frac{EA}{l}$$

➤ Se supone $v_i = 1$
 $u_i, u_j, v_j, \theta_i, \theta_j = 0$



Se puede demostrar calculando por Método de las Fuerzas que para un Desplazamiento Transversal en el extremo i, los esfuerzos en el sistema son:

$$V_i = \frac{12EJ}{l^3}$$

$$M_i = \frac{6EJ}{l^2}$$

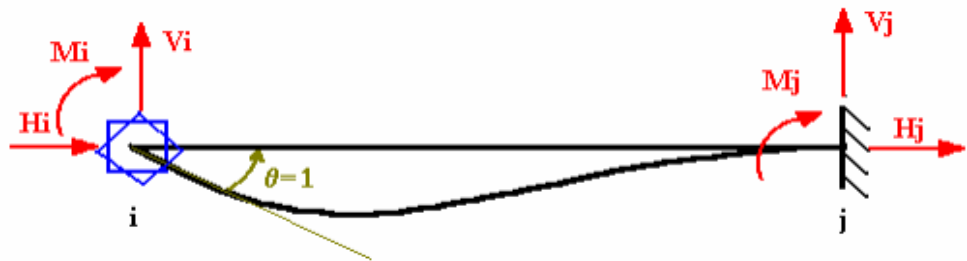
$$V_j = -\frac{12EJ}{l^3}$$

$$M_j = \frac{6EJ}{l^2}$$

$$H_i = 0$$

$$H_j = 0$$

$$u_i, u_j, v_i, v_j, \theta_j = 0$$



De la misma manera que en el caso anterior, tenemos que:

$$V_i = -\frac{6EJ}{l^2}$$

$$M_i = \frac{4EJ}{l}$$

$$V_j = \frac{6EJ}{l^2}$$

$$M_j = \frac{2EJ}{l}$$

$$H_i = 0$$

$$H_j = 0$$

Procediendo de forma análoga para los desplazamientos del Nodo j , obtendremos los restantes coeficientes de la Matriz Rigidez del Elemento.

$$\mathbf{k} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

MATERIALES Y MÉTODOS

2.1. METODOLOGÍA DE LA INVESTIGACIÓN

2.1.1 TIPO DE INVESTIGACIÓN

El nivel de la investigación es de tipo descriptiva, donde se identificaron las características de la población objetivo, necesidades educativas del análisis estructural, se definieron los requisitos y características finales del software desarrollado; además, se recabó el material teórico y metodológico para su posterior análisis e incorporación en el software; y Aplicativa, pues tiene por finalidad producir un software educativo como herramienta tecnológica que incida en manera precisa el uso de este método.

2.1.2 DISEÑO DE INVESTIGACIÓN

El diseño de la aplicación es de campo, debido a que los datos de interés se recogieron de manera directa del entorno donde se dicta la asignatura, entrando en contacto con los estudiantes y profesores de la carrera de ingeniería. El objeto del diseño de la investigación es proporcionar un modelo de verificación que permita contrastar los hechos con la teoría y desarrollar una estrategia o plan general que determine las operaciones necesarias para hacerlo (Sabino, 1995).

2.1.3 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Observación: Consiste en el uso sistemático de los sentidos orientados a la captación de la realidad que se desea estudiar, de la cual se puede obtener información, aun cuando no se quiera proporcionar.

Encuesta: Porque es una técnica anónima y garantiza credibilidad a la investigación, la encuesta va dirigida al alumno de pregrado de la escuela de Ing. Civil de la Universidad Cesar Vallejo – sede Chiclayo.

TÉCNICAS DE ANÁLISIS DE LOS DATOS

Con la información recopilada, se sometió luego al análisis e interpretación de los datos allí reflejados.

Los resultados luego fueron representados a través de gráficos, para observarlos mejor y facilitar la comprensión del análisis.

2.1.4. POBLACIÓN Y MUESTRA

POBLACIÓN:

La población a la cual estaba dirigida la investigación, son trescientos cinco alumnos del pre - grado de la escuela de ingeniería civil 2016 – II de la Universidad Cesar Vallejo sede Chiclayo. Se escogió esta población por su proximidad geográfica a donde se presume dar a conocer la primera versión del software.

MUESTRA

Por ser la variable en estudio categórica nominal y de una pequeña población, el tamaño de la muestra se calcula con la fórmula:

$$n = \frac{k^2 * p * q * N}{e^2 * (N - 1) + k^2 * p * q}$$

Donde,

N = 305 alumnos de pre grado de la escuela de ingeniería civil

p= 0.5 (Probabilidad de éxito)

q= 0.5 (Probabilidad de fracaso)

K =1.96 para el 95% (Factor de confiabilidad)

e= 5% es el error de muestreo

Resulta:

n=170 (tamaño de muestra)

2.2. DISEÑO METODOLÓGICO DE LA PROPUESTA

El diseño metodológico de Galvis Álvaro fue el utilizado para el desarrollo de la propuesta, siendo el diseño expuesto en su libro Ingeniería del Software Educativo (1992).

El diseño metodológico está dividido en cuatro fases, presentadas a continuación:

Fase I. Análisis de Necesidades Educativas.

- 1.1 Análisis de necesidades educativas
 - 1.1.1 Necesidades detectadas.
- 1.2 Análisis de las posibles causas de los problemas planteados.
 - 1.2.1 Solución a las necesidades detectadas.
 - 1.2.2 Carencia de conocimientos del tema a buscar.
 - 1.2.3 Falta de importancia del tema a buscar.
- 1.3 Análisis de las alternativas de solución.
- 1.4 Establecer el papel del computador.
- 1.5 Selección o planeación del desarrollo del Material Educativo.
 - 1.5.1 Recursos disponibles.
- 1.6 Análisis de factibilidad.

Fase II. Diseño del Material Educativo.

- 2.1 Entorno para el diseño del Material Educativo.
 - 2.1.1 Necesidades educativas.
 - 2.1.2 Herramientas a utilizar.
- 2.2. Diseño educativo del Material.
 - 2.2.1 Determinar lo que se espera aprender y lo que se sabe.
 - 2.2.2 Evaluar si se está logrando el aprendizaje.
- 2.3. Diseño del Sistema de Comunicación
- 2.4. Diseño Computacional

Fase III. Desarrollo del Material Educativo.

3.1 Desarrollo y documentación del Material Educativo.

3.2 Revisión del Material Educativo mediante un juicio de expertos.

3.2.1 Verificar la calidad del diseño.

3.2.2 Comprobar si cumple a cabalidad con los objetivos, contenidos y tratamiento de la necesidad que pretende satisfacer el Material Educativo.

Fase IV. Prueba Piloto del Material Educativo.

4.1 Esta fase será la última a efectuarse, donde sólo se involucrará las pruebas y se probará el tutorial con estudiantes interesados.

2.3. HIPÓTESIS

El diseño y desarrollo de software educativo acerca del análisis estructural de vigas continuas por el método de elementos finitos permitirá disminuir la carencia de herramientas tecnológicas que incidan positivamente en el aprendizaje del alumno de pregrado de la Universidad Cesar Vallejo, así como eliminar las deficiencias que se comenten en el uso del método de elementos finitos para el desarrollo de casos específicos sobre análisis estructural.

2.4. VARIABLES

Variable Independiente:

Y1= Diseño y desarrollo de software educativo

Variables Dependientes:

VARIABLES	INDICADORES	VALORES FINALES	TIPO DE VARIABLE
Diseño y desarrollo de software didáctico (Variable independiente)	Empatía con los procesos Fiabilidad de los resultados	Bueno (Cuando están más del 80% satisfechos) Regular (Cuando están entre el 79% y 70% satisfechos) Malo (Cuando están por debajo del 70% satisfechos)	Cualitativa Nominal
Carencias de software del MEF como herramienta de aprendizaje. (Variable dependiente)	Contenido	Completo Regular	Cualitativa Nominal
Deficiencias en el uso del método de elementos finitos. (Variable dependiente)	El proceso y los resultados evidencian el correcto método de los elementos finitos	Totalmente Parcialmente Escasamente	Cualitativa Nominal
Comportamiento de aprendizaje. (Variable dependiente)	Capacidad de respuesta (Aprendizaje)	Satisfecha Insatisfecha	Cualitativa Nominal

TABLA 3: Descripción de variables dependientes

Del Problema

Y2=Comportamiento al aprendizaje.

X1=Carencias de software del MEF como herramientas educativas.

X1=Deficiencias en el uso del método de elementos finitos.

2.5. FACTIBILIDAD TÉCNICA, OPERATIVA, ECONÓMICA Y PRESUPUESTO DE LA INVESTIGACIÓN.

FACTIBILIDAD TÉCNICA

Los equipos necesarios para hacer el trabajo de investigación fueron: una Laptop Advance, con el acceso a Internet, impresora, papelería y material de oficina. Además, el poseer todos los paquetes (software) necesarios para llevar a cabo la investigación. Entre los paquetes usados están: Windows Seven, Office 2013 y Netbeans.

FACTIBILIDAD OPERATIVA

El desarrollo de la investigación fue realizado por los desarrolladores de este proyecto.

Desde el análisis hasta el desarrollo de la propuesta.

Se pidió la colaboración de algunos estudiantes en llenar una encuesta preparada para determinar las necesidades y deseos de los mismos con respecto al tema en estudio.

Los ingenieros tutores de la especialidad de estructuras orientaron el desarrollo de la investigación para que se cumpliera los requerimientos exigidos por el método.

FACTIBILIDAD ECONÓMICA

La investigación generó gastos como las fotocopias de material a ser usado durante la investigación, papel carta para realizar las impresiones, los cartuchos de tinta (color y negro) usados, el gasto en energía eléctrica.

Finalizando con los gastos de encuadernación para hacer entrega del trabajo.

PRESUPUESTO DE LA INVESTIGACIÓN

En el proceso de la investigación ha generado algunos gastos asumidos por los desarrolladores de este proyecto, los cuales detallamos los más relevantes.

CANTIDAD	DESCRIPCIÓN	PRECIO UNITARIO	TOTAL
12	Servicio de luz	10	120
3	Medio millar de papel bond A4	10	30
2	Cartucho negro	5	10
1	Cartucho de color	5	5
1000	Fotocopias	0.05	50
3	Encuadernación	15	45
20	Folder con faster	1	20
48	Pasajes de la ciudad a la Universidad Cesar Vallejo	1.2	57.6
Total			337.6

Tabla 4: Presupuesto de la investigación

2.6. RESULTADOS DEL DIAGNÓSTICO

ANÁLISIS Y PRESENTACIÓN DE LOS RESULTADOS

Entrevista, realizada al alumnado de pregrado de la escuela de Ing. Civil de la Universidad Cesar Vallejo

– sede Chiclayo.

Ítem Nro.	Pregunta
1	¿Reconoce la importancia del método de elementos finitos para el análisis estructural?
2	¿A qué nivel de detalle conoce el método de elementos finitos para el análisis estructural?
3	¿Cree usted que el nivel de aprendizaje es mejor usando material impreso (libros) que haciendo uso de material educativo computarizado?
4	¿Cree usted que la mayoría de software relacionado a la ingeniería no demuestra un proceso detallado del análisis en sus resultados, incidiendo negativamente en la capacidad misma del alumno?
5	¿A utilizado algún software educativo específicamente que permita comprender al detalle el análisis estructural?
6	¿Si contase con una herramienta tecnológica educativa sobre el análisis estructural, usted la usaría?

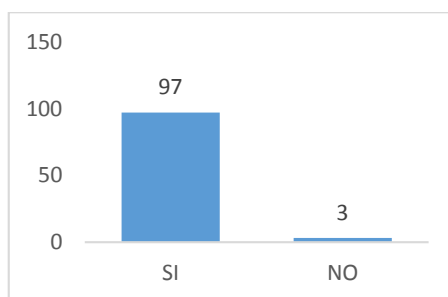
Tabla 5: Preguntas de diagnóstico

ÍTEM No 1: ¿Reconoce la importancia del método de elementos finitos para el análisis estructural?

Tabla Nº 1

Categoría	Fi	Porcentaje
Si	165	97%
No	5	3%
Total	170	100%

Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015



Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015

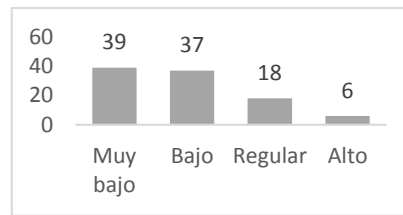
De los alumnos encuestados, el 97% reconoce la importancia de método de elementos finitos para el análisis estructural; sin embargo, el 3% no considera la importancia del uso de este método.

ÍTEM No 2: ¿A qué nivel de detalle conoce el método de elementos finitos para el análisis estructural?

Tabla Nº 2

Categoría	Fi	Porcentaje
Muy Bajo	67	39%
Bajo	63	37%
Regular	30	18%
Alto	10	6%
Total	170	100%

Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015



Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015

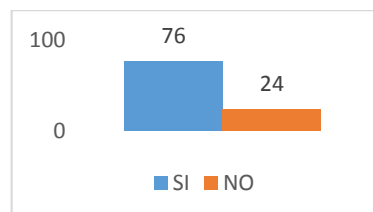
De los alumnos encuestados, sobre el nivel de conocimiento acerca de los detalles del método de elementos finitos para el análisis de estructura bidimensionales posee un nivel alto sólo el 6% , a nivel regular un 18%, un nivel bajo el 37% y a nivel muy bajo el 39%.

ÍTEM No 3: ¿Cree usted que el nivel de aprendizaje es mejor usando material impreso (libros) que haciendo uso de material educativo computarizado?

Tabla Nº 3

Categoría	Fi	Porcentaje
No	150	76%
Si	20	24%
Total	170	100%

Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015



Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015

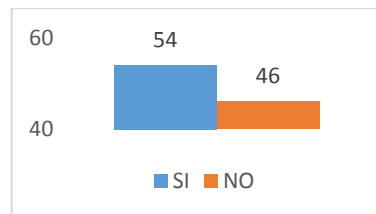
De los alumnos encuestados, el 76% de los encuestados considera que el software educativo incrementa el nivel de aprendizaje; sin embargo, el 24% considera que el nivel de aprendizaje es mejor haciendo uso del material impreso (libros).

ÍTEM No 4: ¿Cree usted que la mayoría de software relacionado a la ingeniería no demuestra un proceso detallado del análisis de sus resultados, incidiendo negativamente en la capacidad misma del alumno?

Tabla N° 4

Categoría	Fi	Porcentaje
Si	92	54%
No	78	46%
Total	170	100%

Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015



Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015

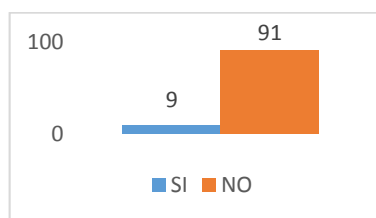
De los alumnos encuestados, el 54% de los encuestados considera que la mayoría de software relacionado a la ingeniería no procesa al detalla el análisis de sus resultados incidiendo negativamente en la capacidad misma del alumnado, sin embargo, el 46% no considera la incidencia negativa.

ÍTEM No 5: ¿Ha utilizado algún software específicamente que permitan comprender al detalle el análisis estructural bidimensional?

Tabla N° 5

Categoría	Fi	Porcentaje
Si	15	9%
No	155	91%
Total	170	100%

Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015



Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015

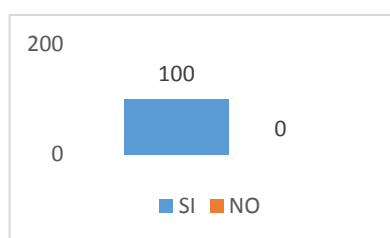
De los alumnos encuestados, sólo el 9% ha podido acceder a un software educativo que permita comprender a nivel medio el análisis estructural bidimensional, sin embargo, el 81% no ha podido hacerlo.

ÍTEM No 6: ¿Si contase con una herramienta tecnológica educativa sobre el análisis estructural, usted la usaría?

Tabla N° 6

Categoría	Fi	Porcentaje
Si	170	100%
No	0	0%
Total	170	100%

Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015



Fuente: Exebio Tantarico y Aguilar Nevado, Encuesta 2015

De los alumnos encuestados, el 100% si utilizaría el software que detalle el análisis estructural bidimensional.

INTERPRETACIÓN DE LOS RESULTADOS

Según los resultados adquiridos de la encuesta, se concluye:

El alumnado reconoce la importancia del método de elementos finitos para el análisis estructural, y admite la necesidad de contar con un recurso educativo digital que describa de manera práctica y detallada el uso de este método.

Los estudiantes desconocen de software que permitan comprender al detalle el análisis estructural; por lo contrario, existe software que tienen la característica de ser precisos en sus cálculos, pero no son los más idóneos en términos de facilidad de aprendizaje por parte del alumnado.

Los estudiantes encuestados manifestaron agrado a la idea de contar con un material extracurricular como recurso educativo computarizado para el análisis estructural.

Por lo expuesto anteriormente, se creó un Material Educativo Computarizado (Software) en la enseñanza – aprendizaje del método de elementos finitos para el análisis estructural; dicho software ofrece información gráfica correspondiente al MEF para el análisis de estructuras e interactúa con el usuario.

2.7. PROPUESTA DE DESARROLLO

DESCRIPCIÓN DE LA PROPUESTA

Nombre del Sistema:

El sistema en desarrollo para el proyecto de grado, tiene por nombre SAEMEF – Sistema de Análisis Estructural por el Método de Elementos Finitos.

El Sistema planteado es un software gratuito conocido usualmente como freeware, es decir una herramienta de libre distribución, pero no de modificación, donde el código fuente estará efecto a propósitos ilustrativos o educativos, ofreciendo las aplicaciones ya compiladas y listas para usarse.

Descripción del sistema.

El sistema está hecho para el aprendizaje – enseñanza del método de elementos finitos sobre el análisis estructural. Los usuarios podrán acceder desde cualquier PC que cuente con el Java Virtual Machine (JVM) de recientes versiones instalado.

El material está dividido en partes, los cuales tienen un contenido instruccional sobre el MEF para el análisis estructural. Al finalizar cada una de las partes se presenta una evaluación donde el usuario podrá revisar y determinar el logro aprendido.

Otro aspecto que presenta el software es permitir el diseño de cualquier caso práctico en la cual se le permite al aprendiz definir una estructura incluyendo: las barras, tipos de soporte y cargas lineales como también puntuales, permitiendo luego generar el cálculo de los momentos y fuerzas cortantes al diseño a manera de simulación, presentando a la vez las matrices locales, globales y las distintas operaciones efectuadas al logro de los resultados por el método de elementos finitos.

El SAEMEF, MEC acerca del MEF para el análisis estructural es sencillo de manejar, la cual anexa un manual que permite al usuario conocer acerca del entorno y uso del mismo software.

POBLACIÓN OBJETO O BENEFICIARIOS

Los beneficiarios del SAEMEF, son los estudiantes de pregrado de la Universidad Cesar Vallejo – Sede Chiclayo, interesados en el aprendizaje de método de elementos finitos para el análisis estructural. Sin embargo, por estar alojado en un servidor gratuito, dicho material podrá ser descargado y utilizado por cualquier persona.

LOCALIZACIÓN

El SAEMEF está alojado en un servidor gratuito MEGA para su descarga:

<https://mega.nz/#!WAdOnR6A>

2.8. PLAN DE ACTIVIDADES

No	Actividad	Observación
1.	Consulta de fuentes de información para identificar el problema	Se indagó por la Internet
2.	Análisis de las causas del problema	Analizar encuestas aplicadas
3.	Análisis de las alternativas de solución	
4.	Determinar el papel desempeñado por el computador en el proyecto	
5.	Planear el desarrollo del MEC (Material Educativo Computarizado)	Aplicando la Metodología de Álvaro Galvis, para la creación de Material Educativo Computarizado, se procedió a realizar todas estas actividades.
6.	Diseñar el entorno del MEC	
7.	Diseño educativo del MEC	

8.	Diseño de comunicación del MEC	
9.	Diseño computacional del MEC	
10.	Desarrollo del MEC	
11.	Revisión del MEC por juicio de expertos	
12.	Revisión uno a uno con usuarios representativos	
13.	Prueba Piloto	

Tabla 6: Plan de actividades

2.9. METODOLOGÍA PARA EL DESARROLLO DE LAS ACTIVIDADES

Para el desarrollo del Material Educativo Computarizado se utilizó la metodología expuesta Álvaro Galvis (1992) en Ingeniería del Software Educativo.

Con esta metodología como la propuesta para el desarrollo del MEC, se precisa en primer lugar determinar las situaciones problemáticas, sus posibles causas y alternativas de solución. Al utilizar esta metodología se conservaron las etapas de un proceso sistemático para el desarrollo de materiales educativos: análisis, diseño, desarrollo, prueba y ajustes e implementación. (Ilustración 13)

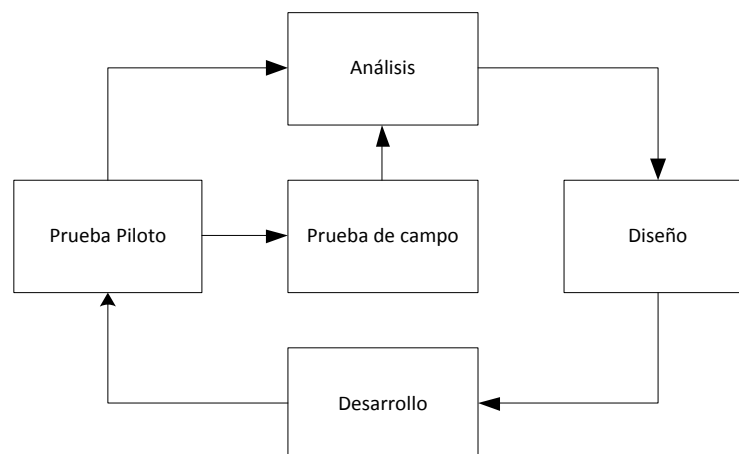


Ilustración 13

Como complemento se realizó el análisis y diseño aplicando UML (Unified modeling Language, lenguaje unificado de construcción de modelos) el cual permite especificar, visualizar, construir y documentar el software que se propone desarrollar. De éste método se elaboró:

- **Modelado de Casos de uso**
- **Diagrama de secuencia**
- **Diagramas de estado**
- **Diagrama de actividades**
- **Diagrama de replieque**
- **Diagrama de clases**
- **Diagrama de paquetes**

FASE I: ANÁLISIS DE LAS NECESIDADES EDUCATIVAS

Dentro del proceso de esta fase en la que se analice la conveniencia de utilizar el computador y de qué manera para que favorezca el aprendizaje de contenido educativo.

Analizar la necesidad educativa conlleva a discernir sobre lo que existe (realidad) y lo ideal (debe ser). En este proceso se plantearon tres interrogantes, a lo que se dio la siguiente respuesta:

¿Qué es lo ideal? (aprendizaje esperado)

Lograr un nivel de conocimiento necesario donde el estudiante comprenda a cabalidad el método de elementos finitos para el análisis estructural.

¿Con lo que existe se puede lograr el conocimiento adecuado acerca del método de elementos finitos para el análisis estructural? *(lo que se puede lograr o aprender con los medios y actividades aplicables)*

En la actualidad, los estudiantes de pregrado de la escuela de ingeniería civil de la universidad cesar vallejo, no cuentan con programas específicos que describan de manera didáctica los detalles en el proceso de análisis estructural por el MEF; sin embargo, existen medios de los cuales se logra obtener de forma directa el resultado de un análisis estructural, sin profundizar y despertar el análisis básico y crítico que debe

conllevar el uso de estos métodos que son propias de la ingeniería. Cabe destacar que el MEF es el método más efectivo para el análisis estructural, en comparación a otros métodos que solo sirven para determinados tipos de estructuras, porque además es actualmente el más usado por distintos softwares existentes en campo de la ingeniería.

Además, es difícil conseguir fuera de horas de clase a un especialista que esté a disposición de brindarnos apoyo al uso de este método. Por lo que no se ha logrado satisfacer esta necesidad educativa.

¿Qué falta por alcanzar? (aprendizaje esperado)

El material educativo computarizado (MEC) que sirva de material autoinstruccional para el aprendizaje del MEF para el análisis estructural.

Necesidades Detectadas

Existen diferentes tipos de necesidades educativas, según lo señalan Burton y Merrill.

1. Necesidad normativa: se toma como ideal una norma o patrón existente, se determina en qué medida la población objeto del sistema educativo alcanza dicho estándar y se establece la diferencia con el patrón.

2. Necesidad sentida: son sinónimo de un deseo de saber algo. Se identifica simplemente preguntándole a la gente qué quiere aprender.

3. Necesidad expresada o demanda: es similar a la idea económica de que, si la gente necesita algo, lo solicitará.

4. Necesidad futura o anticipada: resulta de prever las necesidades que se demandarán en el futuro, con base en el seguimiento a los planes de desarrollo relacionados, así como a los avances científicos y tecnológicos.

Teniendo en cuenta lo anterior, se considera las necesidades que permiten buscar una solución al problema planteado en el Capítulo I, son:

Necesidades sentidas y reflejadas por los estudiantes en la encuesta explícita en el Capítulo IV, de contar con una herramienta tecnológica que facilite el aprendizaje al detalle sobre el MEF para el análisis estructural bidimensional; asimismo se observan necesidades futuras ya que este método de elementos finitos se viene aplicando cada vez con mayor énfasis en los distintos softwares en el campo de la ingeniería, pues es la más cercana y exacta para distintos diseños estructurales.

Para analizar las necesidades educativas también se usó una metodología representada en el siguiente esquema (Ilustración 14):

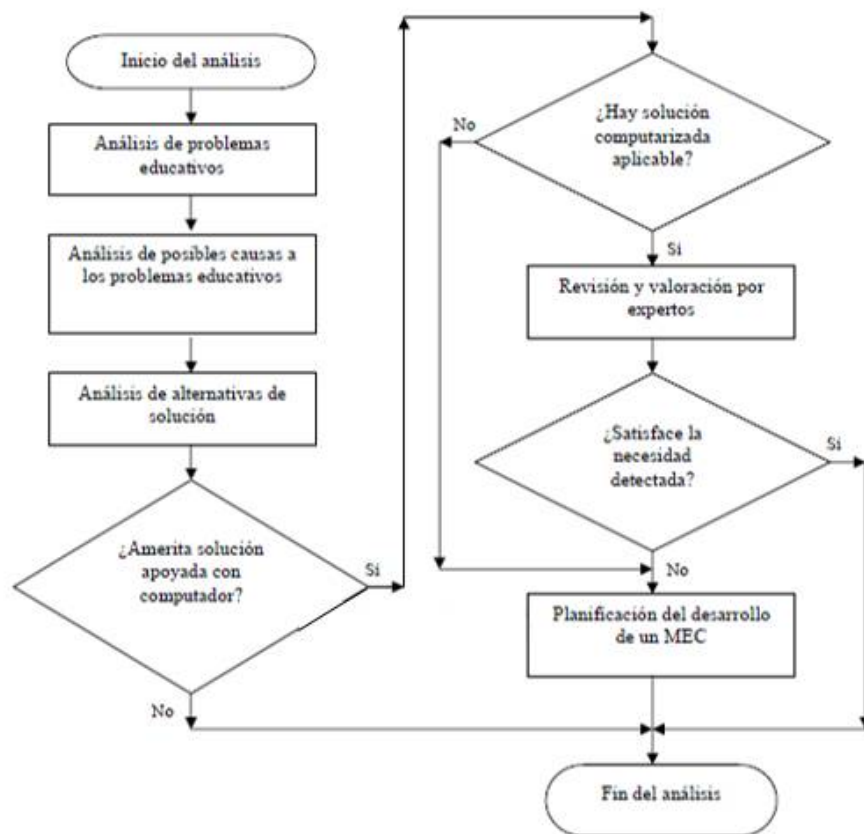


Ilustración 14

Una vez identificadas las necesidades, se analizaron los problemas educativos que se presentan en nuestro caso de estudio.

Análisis de Problemas Educativos.

Se detectaron algunos problemas educativos; unos de operatividad y otros de aprendizaje, señalados a continuación:

Dificultad para conseguir medios humanos que impartan el aprendizaje requerido.

La falta de una aplicación idónea para el aprendizaje del MEF al momento de generar pruebas de análisis de diferentes casos estructuras bidimensionales.

El material educativo es en su mayoría práctico, por lo que es necesario comprobar lo dicho en el material visual o textual.

Con la metodología conductista se comprendió la selección del medio de instrucción tomando en cuenta el estímulo que se deseó aplicar en la enseñanza. Por lo que se recomendó en particular, el uso del computador, ya que se desea una interacción entre el sujeto y el objeto de aprendizaje o la fuente de conocimiento.

Cabe señalar que esto no es una condición suficiente para seleccionar apoyo computarizado, ya que otros medios como textos, material audiovisual, entre otros; pueden suplir este tipo de estímulo.

Sin embargo, el MEF si bien es un modo ordenado y sistemático de proceder para llegar al resultado de obtener los momentos y fuerzas cortantes en una estructura, su mecanismo en la que se centra este método en la de generar las matrices locales y globales como operarlas es complicado, que requiere mucho criterio al definir las matrices como la exactitud al momento de operarlas, hecho que puede suscitar a que se puede producir errores en un desarrollo manual y no computarizado, sobre todo cuando se trata de estructuras bastante complejas. Por lo que, un aprendizaje efectivo de este tipo de métodos es más útil apoyarse con este tipo de herramientas que permita validar los ejercicios acerca del uso de este método para problemas estructurales complejos.

Análisis de Posibles Causas de los Problemas Detectados.

Las posibles causas de los problemas detectados se obtuvieron tras la indagación con los estudiantes, de los cuales se describen:

El material existente sobre el MEF describe de manera compleja y con carencia de simpatía el uso de este método y solo para estructuras básicas.

El tiempo también puede ser una posible causa del fracaso. Un curso presencial, requiere días de dedicación, y muchos estudiantes interesados en aprender no pueden disponer del tiempo por motivos laborales.

La metodología utilizada o los medios en los que se apoya el proceso de enseñanza – aprendizaje en la Universidad Cesar Vallejo, se torna en muchas oportunidades difíciles para el estudio del MEF, por lo que se podría usar el material educativo computarizado como materia de aprendizaje.

Análisis de las alternativas de solución.

El hecho de saber que hay problemas y a qué se deben, no necesariamente nos deducen que la solución sea un apoyo computarizado. Sin embargo, por las características del caso se recomienda este como un medio capaz de subsanar la deficiencia.

La solución al problema planteado es:

Solución académica. Generación de un material educativo computarizado de fácil acceso, de amigable interfaz y sin complejidad que permita al estudiante un autoaprendizaje efectivo. El MEC propicia la solución al problema de motivación por ser un medio atractivo y significativo para el aprendiz; en la que su exploración impulsa adentrarse en el contenido de lo que está aprendiendo.

Establecer el papel del computador.

El papel del computador propicia el apoyo informático para darle solución al problema por medio de:

Un sistema de análisis estructural bidimensional por el método de elementos finitos, la cual permite afianzar de conocimiento acerca de este método a través de un entorno amigable y entretenido.

Además, según las características de esta herramienta, incorpora una aplicación inteligente, que permite el diseño de cualquier caso práctico en la cual el aprendiz puede definir cualquier estructura bidimensional donde incluya: barras, tipos de soporte y cargas lineales como también puntuales; para luego generar el cálculo de los momentos y fuerzas cortantes al diseño a manera de simulación, presentando a la

vez las matrices locales, globales y las distintas operaciones efectuadas al logro de los resultados por el método de elementos finitos, como lo haría un experto en la enseñanza del tema.

Selección o planeación del desarrollo del Material Educativo Computarizado.

En este paso, se determinó si hay una solución que satisfaga la necesidad detectada, que se encuentre ya resuelta o si hay que desarrollar un MEC para esto. Existen muchos MEC's para ser usados, sin embargo, tomando en cuenta variables como: tipo de estudiantes, nivel educativo, área de contenido, y la modalidad de acceso, entre otros; se determinó en la creación de un MEC que satisfaga las necesidades del material educativo computarizado. Sin embargo, se procedió a hacerlo revisar por expertos en contenidos e informática, por el Director de Escuela Ing. Ulises Guevara Paico y el Ing. Marco Polo Rojas respectivamente. Quienes sugirieron que se desarrollara una aplicación con la base de contenidos, casos prácticos y/o la incorporación de un simulador y que este sea a nivel de escritorio como una aplicación de fácil instalación para que se pueda acceder desde cualquier computador

Recursos Disponibles.

Bibliográficos:

- Material bibliográfico obtenido desde la Internet

Humanos:

- Director de Escuela Ing. Ulises Guevara Paico. Experto de Contenido.
- José Luis Aguilar Nevado y José Antonio Exebio Tantarico, expertos informáticos y autores del proyecto.

Análisis de Factibilidad. Por ser una aplicación a nivel de escritorio y de la cual se puede ejecutar en cualquier plataforma donde exista una JVM independiente de su arquitectura permitirá su desarrollo como tener portabilidad necesaria para su uso. Con respecto al contenido, con ayuda del especialista en contenido se seleccionó el contenido para el aprendizaje análisis estructural por el método de elementos finitos, sin ningún tipo de inconveniente.

FASE II: DISEÑO DEL MATERIAL EDUCATIVO

Según los resultados de la fase de análisis se diseña el MEC, estableciendo el: ambiente y actividades de aprendizaje, el sistema de comunicación entre el programa y el usuario, así como las especificaciones funcionales necesarias para el desarrollo del MEC.

ENTORNO PARA EL DISEÑO DEL MEC.

A partir de los resultados del análisis, se hacen explícitos los datos que caracterizan el entorno del material que se va a diseñar: destinatarios, área de contenido, necesidad educativa, limitaciones y recursos para los usuarios del MEC, equipo y soporte lógico que se va a utilizar.

El MEC será utilizado por los estudiantes de nivel intermedio de pregrado cuyas edades están comprendidas entre los 18 y 25 años aproximadamente, los cuales están interesados en aprender los métodos de análisis estructural siendo un requerimiento básico y de importancia en el perfil profesional de egresado.

Debido a que un software es dirigido a la educación superior, los usuarios poseen habilidades de manejo de un sistema operativo como explorar software a nivel de escritorio, que son precisamente los requerimientos mínimos para utilizar el MEC. Y sobre, los conocimientos relevantes para el estudio del tema, muchos de ellos al menos, conocen lo básico de análisis de estructuras, por cursos previos.

El MEC es para ser usado de forma individual. No es necesario la ayuda de un tutor o profesor durante las sesiones de estudio.

Las características mínimas requeridas para los equipos en los que corre el MEC son aquellas plataformas donde exista una JVM para que se puede ejecutar el código objeto independientemente de la máquina en el que se implemente.

Diseño educativo

Contenido

El contenido a desarrollar es:

- Cálculo de Sistema de Ecuaciones Lineales usando matrices
- Características generales de las estructuras
- Método de rigidez
- Desarrollo de casos

Los puntos a desarrollar en el contenido del MEC son principios básicos en lo que a análisis de estructuras bidimensional por el MEF se refiere.

Tipo de ambiente

El problema educativo estudiado indica la necesidad de contar con un MEC que permita al usuario conocer de manera clara el proceso del MEF para el análisis de estructuras bidimensionales. Por lo que, el MEC se ha orientado en un sistema de ejercitación y práctica después de revisar aspectos conceptuales básicos del proceso del MEF, y que sirva de herramienta de autoaprendizaje o como herramienta de apoyo al docente para impartir clases y motivar al alumno.

Evaluación

El objetivo de este software es que sea de carácter reproductivo, es decir, que el aprendiz a través del MEC adquiera conocimiento acerca de los principios básicos en lo que a análisis de estructural bidimensional por el MEF se refiere, una vez que lo comprenda este apto para aplicarla en diversas circunstancias de análisis estructural. Esto conlleva, que al final del recorrido el estudiante sea evaluado para con el fin que el alumno mida su nivel de aprendizaje respecto al MEF, donde la evaluación, contiene preguntas de selección y algunos ejercicios relacionados al análisis estructural.

La Motivación

La motivación es elemental para el éxito del MEC, de no hacerlo resultaría inútil la construcción del software. Para una persona adulta, el ser el software de acceso libre (gratis), es ya una primera motivación.

El software hace gala en una de sus partes de contener un simulador apropiado para mayor aprendizaje del MEF para el análisis estructural bidimensional.

El estudiante al finalizar el proceso de evaluación de manera satisfactoria, el software permite le da la opción de recibir un certificado de aprobación.

DISEÑO DEL SISTEMA DE COMUNICACIÓN

Siendo el hardware y el software con los requerimientos básicos son fundamentales para el funcionamiento del MEC. Es en esta fase del proceso de diseño donde se define la interfaz del software, medio por el cual el usuario va a estar en contacto directo con el programa de ejercitación y práctica. Luego considerando las condiciones de la población al que va dirigido se eligió las características que conllevará el diseño de la interfaz para el MEC (micro mundo), las cuales son:

- Zona de encabezado. En esta zona muestra las posibles navegaciones del usuario dentro de la aplicación.

- Zonas de Control de Programa. Zona en la que el usuario podrá controlar el flujo de su ejercitación y práctica, y por ende el ritmo de ejecución del programa.

- El control de flujo de ejercitación y práctica refiere a la activación de sus diferentes capítulos del MEC a manera de menú.

- El control del ritmo. Tiene que ver con la posibilidad que tiene el usuario de decidir cuándo continuar con el aprendizaje y visualizar las secciones a las que puede recurrir.

- Zona de trabajo. Es aquella donde el usuario tiene a su disposición lo que sirve de base para aprender (teoría, ejemplos, evaluaciones, simulador).

En la siguiente figura se muestra las zonas de comunicación en diseño para el desarrollo del MEC (Ilustración 15):

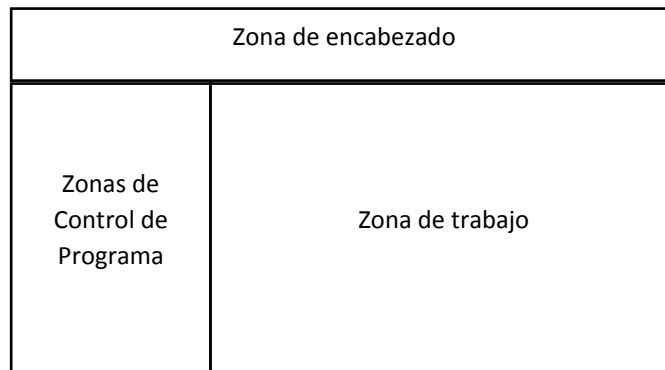


Ilustración 15

La zona de control del programa, se encuentra el menú de cada uno de los capítulos correspondientes al aprendizaje del MEF, con sus respectivas prácticas y ejercicios. Este menú se presente en forma gráfica y textual, y se manipula con el uso del mouse.

Respecto a las características de los elementos gráficos en la interfaz es necesario considerar que los elementos multimedia como el texto, colores e imágenes que definan la estética sean los apropiados para el MEC con el fin de asegurar el éxito de la comunicación entre el usuario y la aplicación. Las características por cada una de ellos son:

Texto.

- Observarse en su totalidad en la pantalla, sea conciso y breve.
- Adecuado tipo, estilo y tamaño de letra.
- Espacio entre las líneas, de manera que no se vea recargada y no canse al lector.

Color.

No se usará gran variedad de colores, ya que no está destinado para un público infantil; si bien se usa para diferenciar niveles, estados y distintos objetos, podemos tener alternativas diferenciadoras como: sombras, patrones o sonidos. Los colores en cuanto a su estado, tienen la función de indicar, por ejemplo: el

color rojo de la cual se puede interpretar como error, verde como una notación adicional y amarillo como especie de advertencia.

Imagen.

En cuando a las imágenes podemos distinguir dos tipos de imágenes, las de tipo mapa de bits y vectoriales. Las de tipo mapa de bits se forma esta por un conjunto de bits de diferente color, las cuales van formando en conjunto la imagen; y las de tipo vectorial están formadas por líneas y/o figuras geométricas de valores numéricos que en conjunto forman la imagen requerida, siendo esta última con la cual mayormente se va a trabajar ya que se trata de elementos estructurales de ingeniería.

Características muy importantes para el correcto aprendizaje del MEF, son la de consistencia de la imagen relacionada a la ingeniería estructural bidimensional y además sea de buena calidad para que tome interés por un público de formación superior.

DISEÑO COMPUTACIONAL

En el diseño computacional figura las funciones del MEC para cada uno de los tipos de usuarios y especificando adicionalmente la manera que el sistema efectuará las acciones para lograr satisfacer las necesidades, tanto en el nivel educativo como en el nivel computacional, a través de la estructura lógica y la estructura de datos respectivamente.

Toda esta información ayuda a redefinir el modelo antes de iniciar la fase de desarrollo. Además, permite validar si el modelo es completo y permite satisfacer todos los requerimientos de la aplicación. Una vez cumplidos los puntos del diseño se procede a la parte del desarrollo.

• Funciones para los tipos de usuario alumno y/o tutor

Funciones de apoyo para el alumno.

- Permitir controlar el ritmo y la secuencia del aprendizaje.

- Ofrece una explicación sobre el sistema, si se requiere.
- Presentar teoría, ejercicios y prácticas.
- Permitir demostrar cuánto se domina el tema.
- Lleve registro del avance del usuario.

Funciones de apoyo para al tutor.

- Permitir manejar la secuencia de enseñanza
- Permitir la ejercitación de casos prácticos.
- Consultar resultados obtenidos a nivel de escritorio por alumno.

• **La estructura lógica del software educativo.**

Modelado de casos de uso

Nivel 0

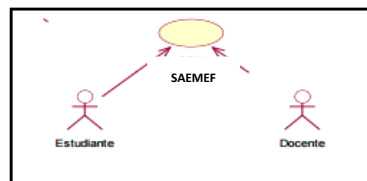


Ilustración 16

Especificación de Casos de Uso: Nivel 0

Descripción: Este caso de uso es la representación de los actores que intervienen en el sistema

Actores: Estudiante, Docente

Nivel 1

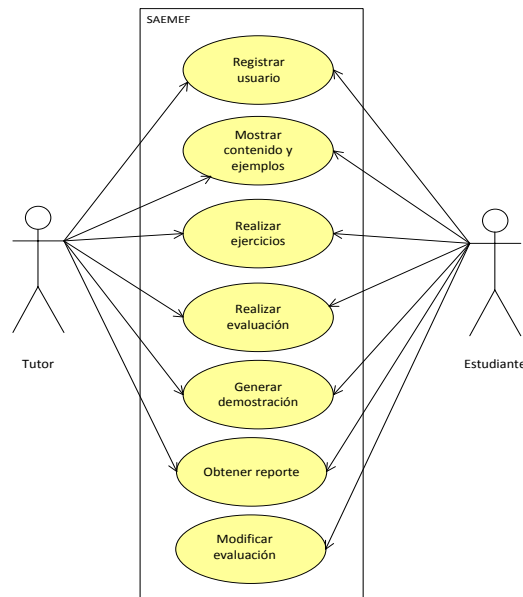


Ilustración 17

Especificación de Casos de Uso: Nivel 1

Descripción: Este caso de uso describe las funciones principales realizadas por los diferentes tipos de usuario en el sistema.

Actores: Estudiante, Docente

A continuación, se ofrece una breve descripción de cada uno de los casos de uso mostrados en el diagrama:

Registrar usuario: Registrar datos del usuario que participa del MEC, datos que servirán para tener registro de su avance.

Mostrar contenido: Mostrar por cada capítulo los detalles específicos correspondientes el análisis estructural bidimensional por el método de la rigidez

Realizar ejercicios: El sistema presenta algunos ejercicios demostrativos para afianzar el contenido y ejemplos que brinda el sistema.

Realizar evaluación: Mediante algunas preguntas acerca del contenido y del MEF el alumno demuestra su aprendizaje. Si el participante logra aprobar la evaluación tiene la opción de imprimir una constancia de su participación.

Generar demostración: El sistema presenta la alternativa que el participante pueda diseñar su estructura bidimensional, para luego mostrar al detalle el análisis estructural por el método de elementos finitos.

Mostrar reporte: Obtener los avances desarrollados por el usuario tras haber participado del sistema de aprendizaje.

Modificar evaluación: El tutor modificar las preguntas de la evaluación de cada capítulo.

Diagramas de secuencia

Después de haber finalizado el modelado de caso de uso, se procedió a realizar los diagramas de secuencias para mostrar la funcionalidad del MEC y el orden de interacción entre los diferentes usuarios (estudiante y docente), con el software educativo SAEMEF.

Diagrama de secuencia (docente)

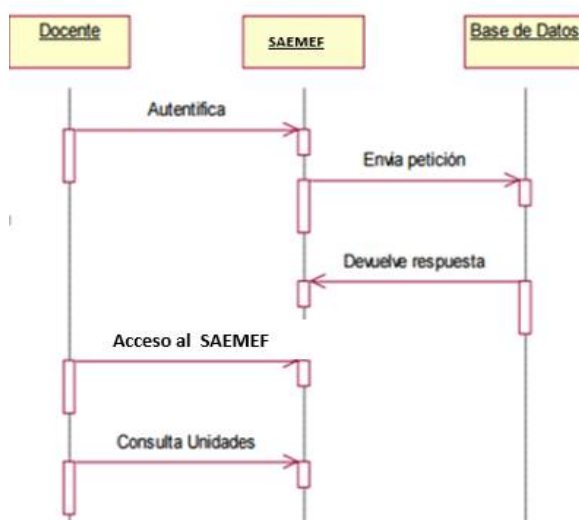


Ilustración 18: Diagrama de secuencia acceso Docente

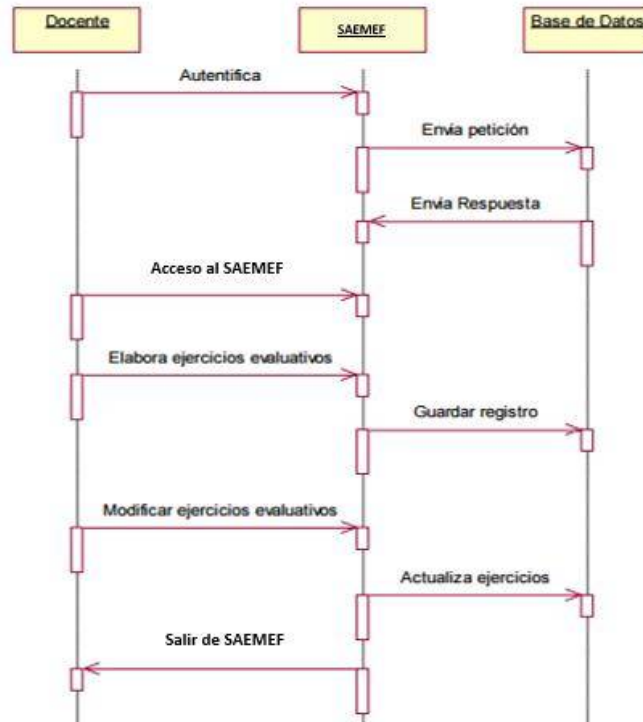


Ilustración 19: Diagrama de secuencia actualizar y modificar ejercicios (Docente)

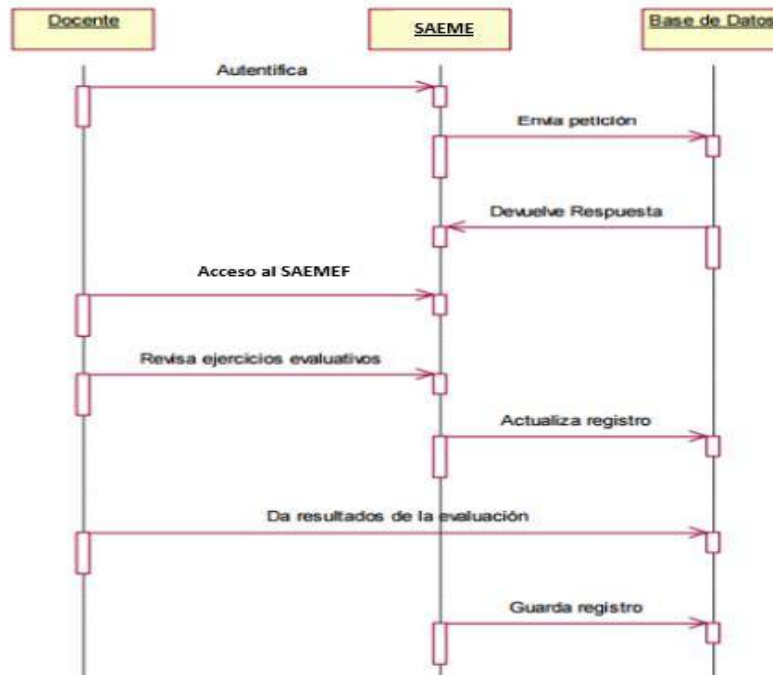


Ilustración 20: Diagrama de revisa respuestas docente

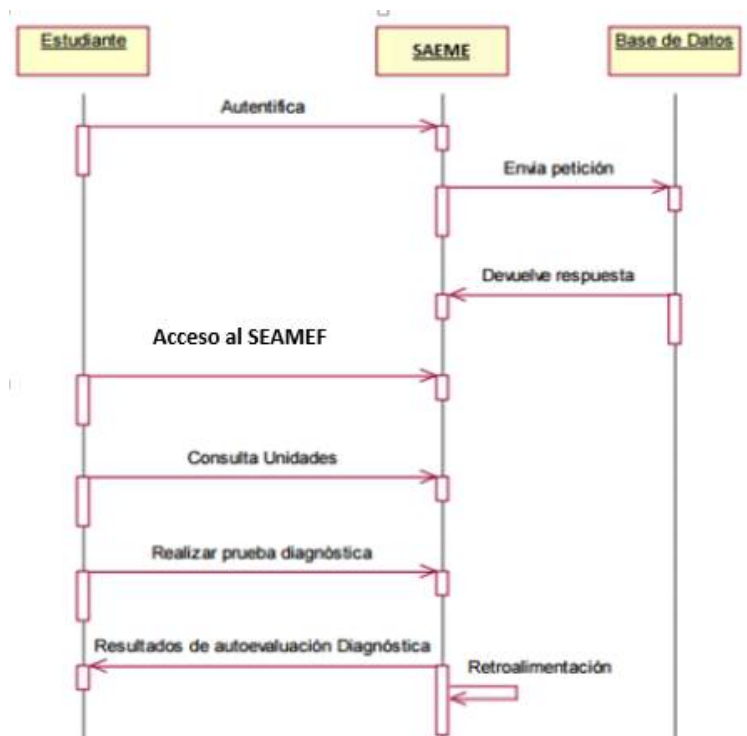


Ilustración 21: Diagrama de secuencia Estudiante (Realizar prueba diagnóstica)

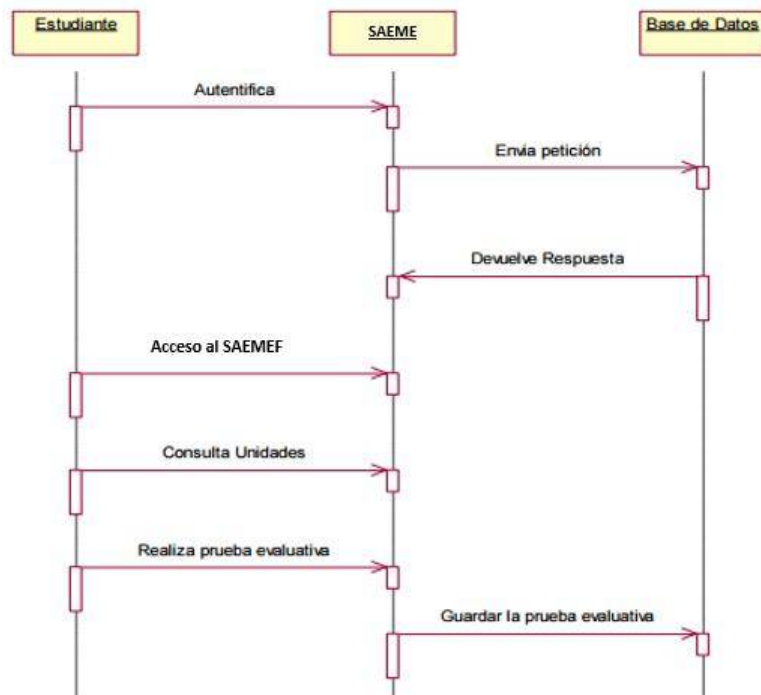


Ilustración 22: Diagrama de secuencia Estudiante (Realizar prueba diagnóstica)

Diagramas de estado

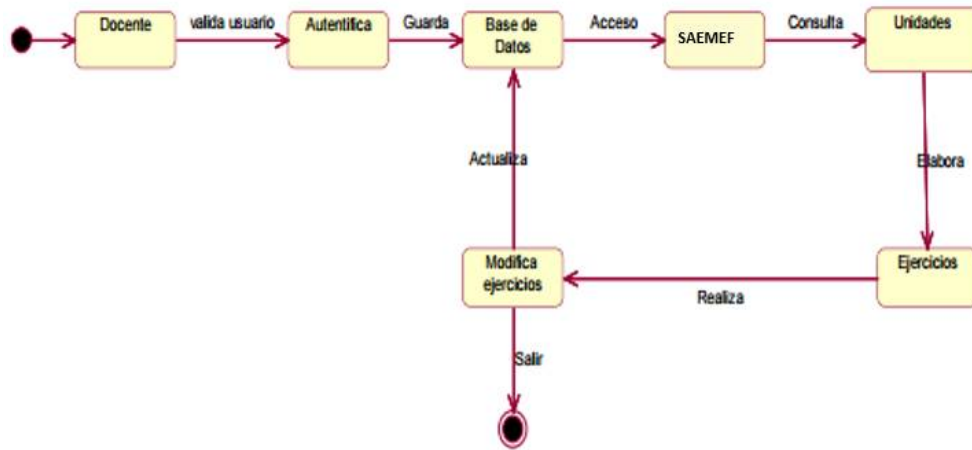


Ilustración 23: Diagrama de estado del docente

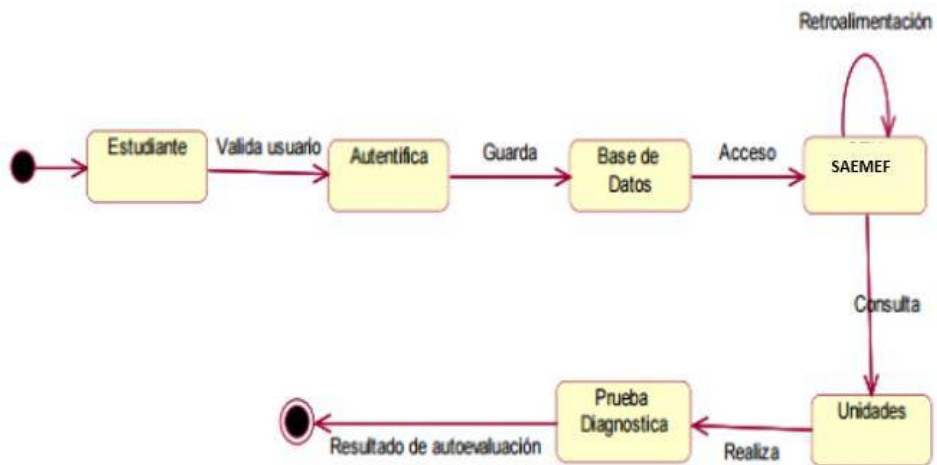


Ilustración 24: Diagrama de estado del estudiante

Diagrama de actividad

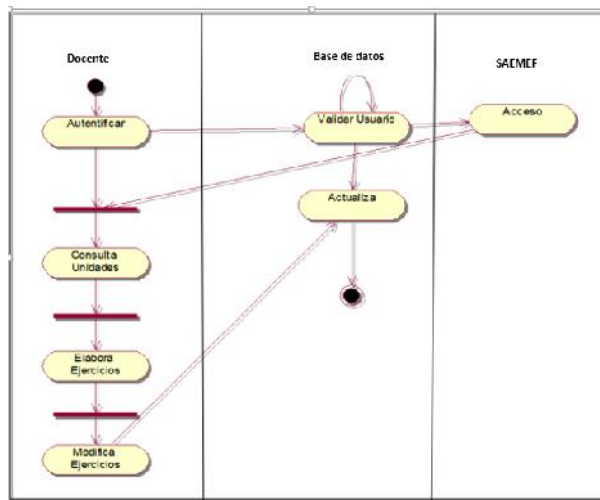


Ilustración 25: Diagrama de actividad de docente

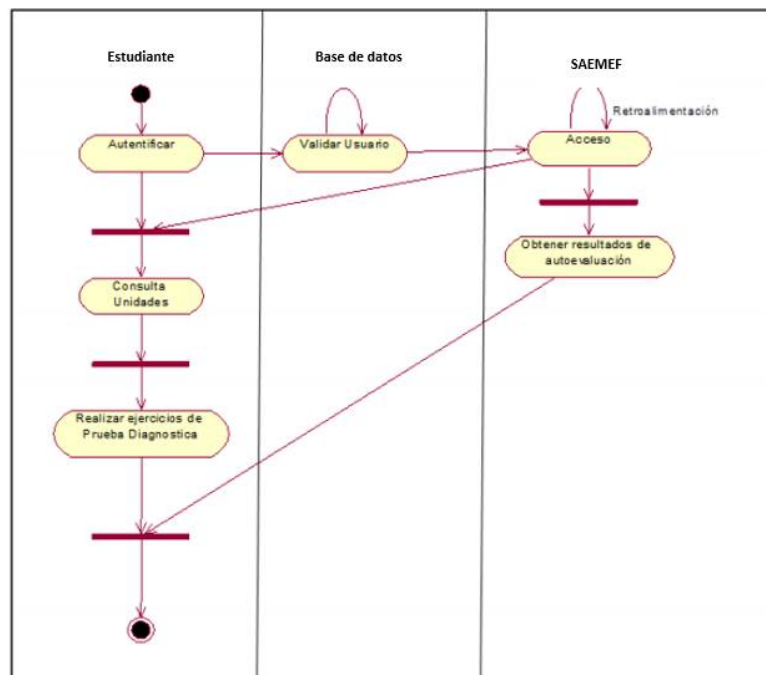


Ilustración 26: Diagrama de actividad de estudiante

Diagrama de despliegue

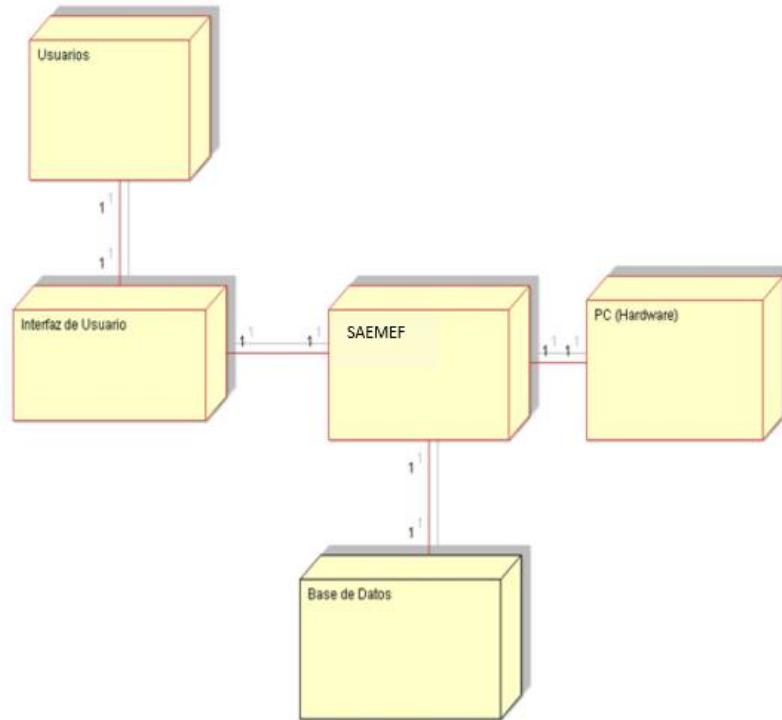


Ilustración 27: Diagrama de despliegue SAEMEF

Diagrama de clases

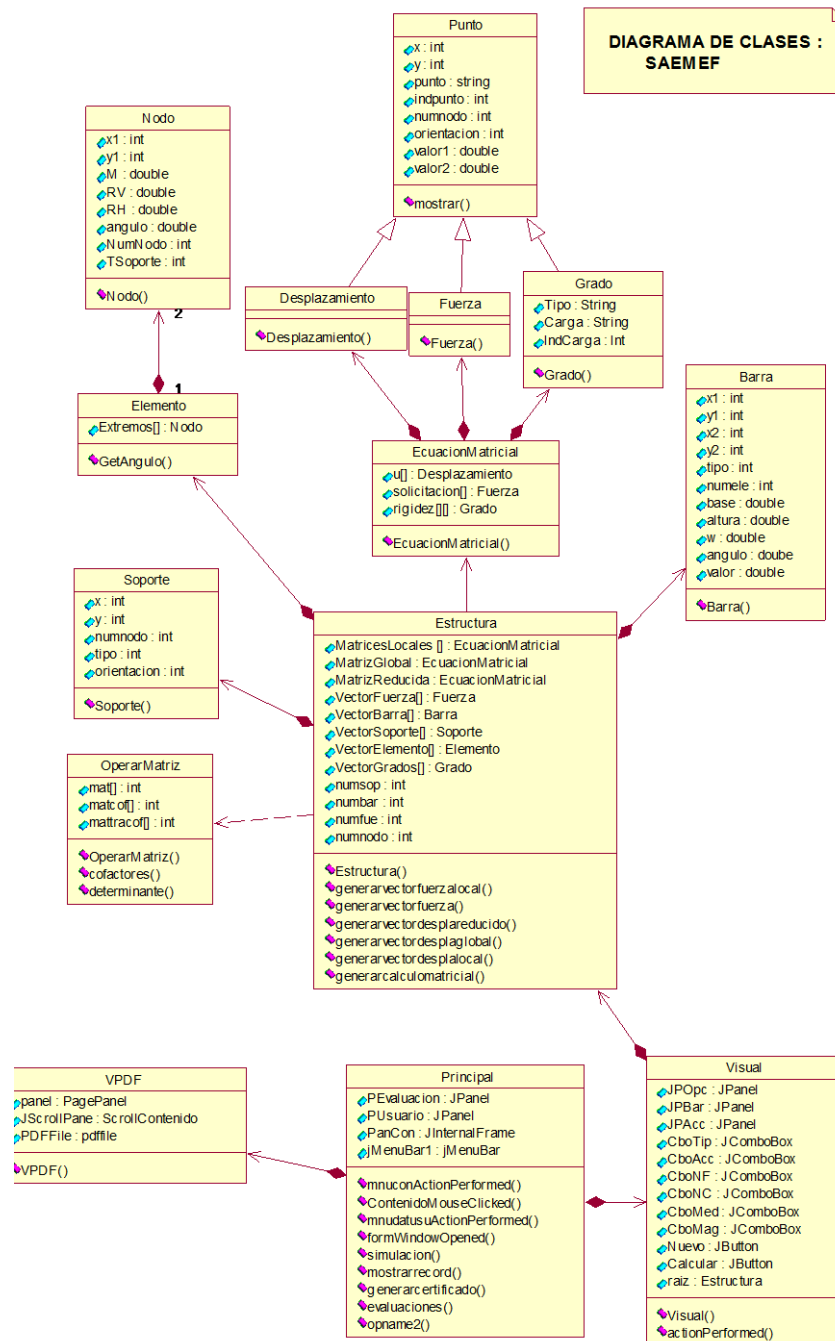


Ilustración 28: Diagrama de clases del SAEMEF (Simulador)

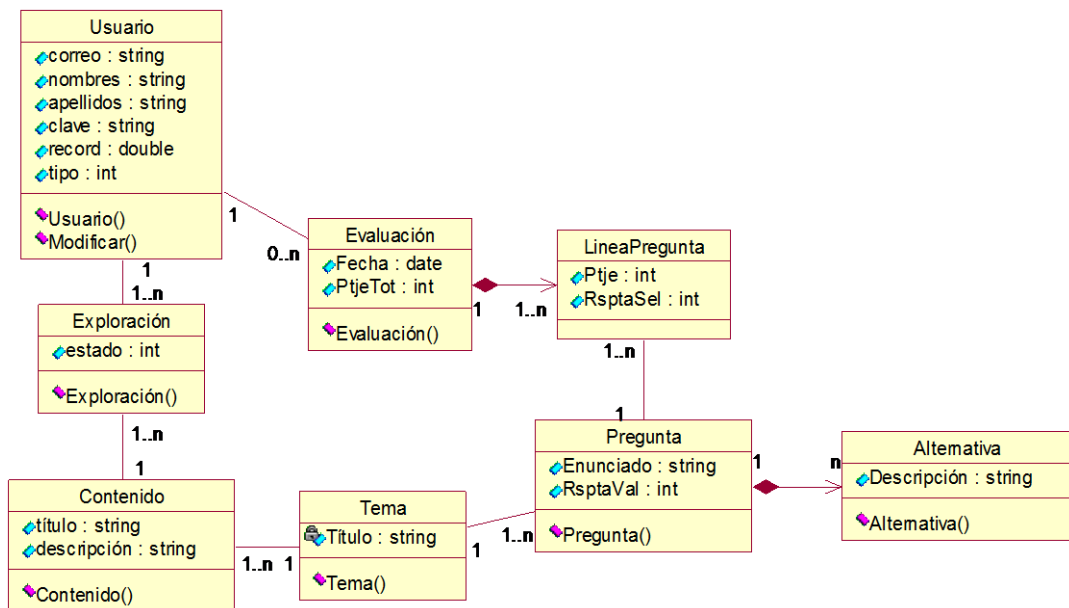


Ilustración 29: Diagrama de clases del SAEMEF (Registro de datos)

Tras haber generado el diagrama de clase para un sistema complejo, se separó el diagrama en secciones que muestren la lógica de la aplicación. Esto se puede hacer físicamente segmentando el diagrama de clase, usando diagramas separados para cada sección, o agruparlos en paquetes del proyecto, así como la lista de clases que compone a cada paquete (Ilustración 18).

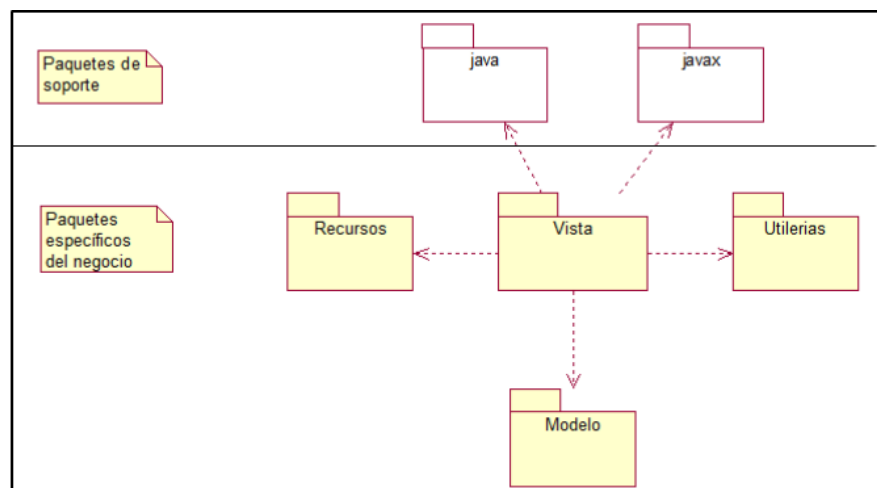


Ilustración 30: Paquetes del SAEMEF

FASE III: DESARROLLO DE APLICACIÓN

En esta fase se desarrolla el material educativo computarizado, en la que según Alvaro Galvis describe el desarrollo aplicación desde cero, o evaluar algún Gestor de contenidos que satisfaga las necesidades educativas con sus características diseñadas en la anterior fase.

De los varios sistemas de gestión de cursos, como: Moodle, Claroline, entre otros; con ventajas en la administración de contenidos como en sus herramientas colaborativas, además de su reconocimiento internacional como entornos e-learning, se ha considerado en el desarrollo de un sistema de escritorio que permita ejecutarse desde cualquier equipo sin la necesidad de estar conectado a internet para ejercitarse y lograr el aprendizaje del método de elementos finitos para el análisis estructural.

Según la arquitectura para el desarrollo del programa, tenemos que determinar el desarrollo de las clases que conformarán el sistema conforme se planteó en el diseño.

Se describe primero los paquetes que se utilizarán:

CalculoMatricial: contiene la clase principal del programa, desde la que se ejecuta el resto del código.

CalculoMatricial.modelo: contiene las clases más abstractas del programa, que forman el modelo de la aplicación.

CalculoMatricial.utilerias: son clases que sirven de soporte al programa, pero que no forman parte del modelo, ni la vista, ni el controlador. Tal es el caso de las validaciones que se hacen para los datos de la tabla de entradas.

CalculoMatricial.Vista: en este paquete se encuentran las clases que forman la vista y el controlador. No se hizo un paquete especial para el controlador, pues en esta aplicación vista y controlador están muy unidos, por lo que sería poco práctico separarlos.

CalculoMatricial.Recursos: en este paquete se encuentran los logos que se usan dentro de la aplicación, contenido del material de aprendizaje, así como el manual de usuario. En la siguiente sección se empezarán a identificar las clases que formarán parte de los paquetes listados anteriormente.

Poco a poco se fueron agregando las clases necesarias para cumplir con los requerimientos, quedando así la organización de las clases en paquetes (ilustración 31):

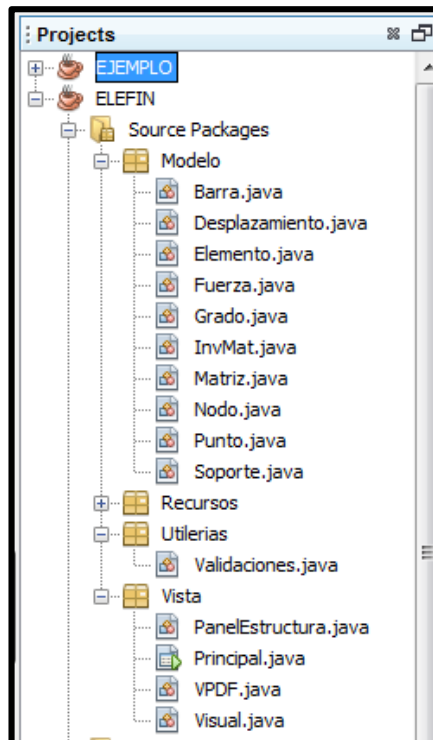


Ilustración 31: Clases en los paquetes de la aplicación.

CalculoMatricial.Vista

Principal.java: es la clase principal del proyecto (contiene el método main). Desde aquí se crea la ventana principal, el cual permite ingresar a la aplicación, es clase heredada de la clase JFrame en la que implementamos una interfaz visual con la librería Swing. En el constructor instanciamos objetos de las clases JMenu, JMenuBar, JMenuItem, JPanel y JFrame; es en estos últimos donde ubicaremos los controles visuales.

PanelEstructura.java: Clase heredada de la clase JPanel al que se le sobrescribe el método paint() para poder dibujar la estructura con sus componentes y en la que implementamos los métodos abstractos en

una interfaz `MouseListener`, `MouseMotionListener`. Conforme se va dibujando los componentes de la estructura se va generando los objetos de las clases importadas del paquete `Modelo`.

VPDF.java: Clase heredada de la clase `JFrame`, la cual apoyado de librerías permitirá incluir en un Panel el acceso a los archivos PDF del contenido para el aprendizaje del método de elementos finitos para el análisis estructural.

Visual.java: Clase definida por código con herencia de la clase `JFrame` e implementando con el método `ActionListener`. Esta clase permitirá armar una ventana que de soporte a al **PanelEstructura.java**.

CalculoMatricial.Modelo

Barra.java: es la clase que permite contener los datos de las vigas ubicadas en los tramos de la estructura.

Soporte.java: es la clase que permite contener los datos del tipo de soporte y donde se encuentra ubicado según la estructura.

Nodo.java: es la clase que permite contener datos del punto donde se encuentra ubicado de toda la estructura.

Elemento.java: es la clase que contiene los detalles de un tramo en la estructura, considerando dos objetos de la clase `nodo` como los extremos del elemento.

Grado.java: es la clase con los atributos correspondientes al grado de libertad.

Punto.java: es la clase abstracta de la cual heredan sus atributos y métodos las clases `Desplazamiento.java` y `Fuerza.java`.

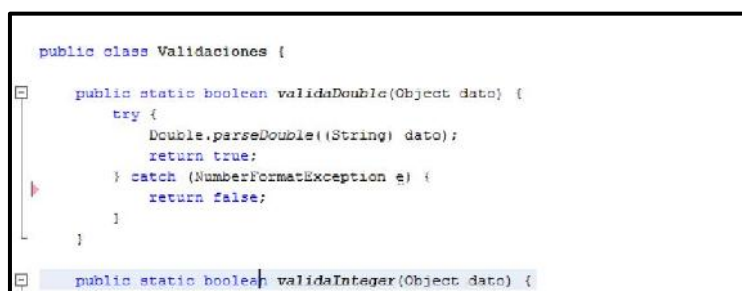
Desplazamiento.java: es la clase donde se registra los desplazamientos de los nodos después de procesado el cálculo matricial. De esta clase se generan los arrays para el desplazamiento global y el desplazamiento reducido de la matriz reducida.

Fuerza.java: es la clase donde se registra las fuerzas de los nodos después de procesado el cálculo matricial. De esta clase se generan los arrays para la fuerza global y la fuerza reducida de la matriz reducida.

Matriz.java: de esta clase se generan los objetos de las matrices locales y la matriz global. Es una clase que contiene arrays de la matriz global como locales según la discretización definida por el usuario para la estructura.

CalculoMatricial.Utilería

Validaciones.java: clase auxiliar para realizar las validaciones de los datos de entrada en la ventana principal (Ilustración 32). Estas validaciones son para comprobar si un dato es un número entero, un número real (double) o si es nulo.



```
public class Validaciones {  
    public static boolean validaDouble(Object dato) {  
        try {  
            Double.parseDouble((String) dato);  
            return true;  
        } catch (NumberFormatException e) {  
            return false;  
        }  
    }  
    public static boolean validaInteger(Object dato) {
```

Ilustración 32: Validación de variables

Teniendo además en cuenta el propósito del MEC de guardar los datos de la interacción del usuario en algunos casos de la aplicación, es decir de guardar los datos de la memoria RAM en una base de datos, es aquí donde se tiene que transformar esta información de objetos a tablas y columnas en una base de datos (Ilustración 33).

Ante la correspondencia entre tablas y clases, para el desarrollo de la misma aplicación se generó la siguiente base de datos con las tablas donde se hace necesario el registro permanente y no solo temporalmente en la memoria RAM.

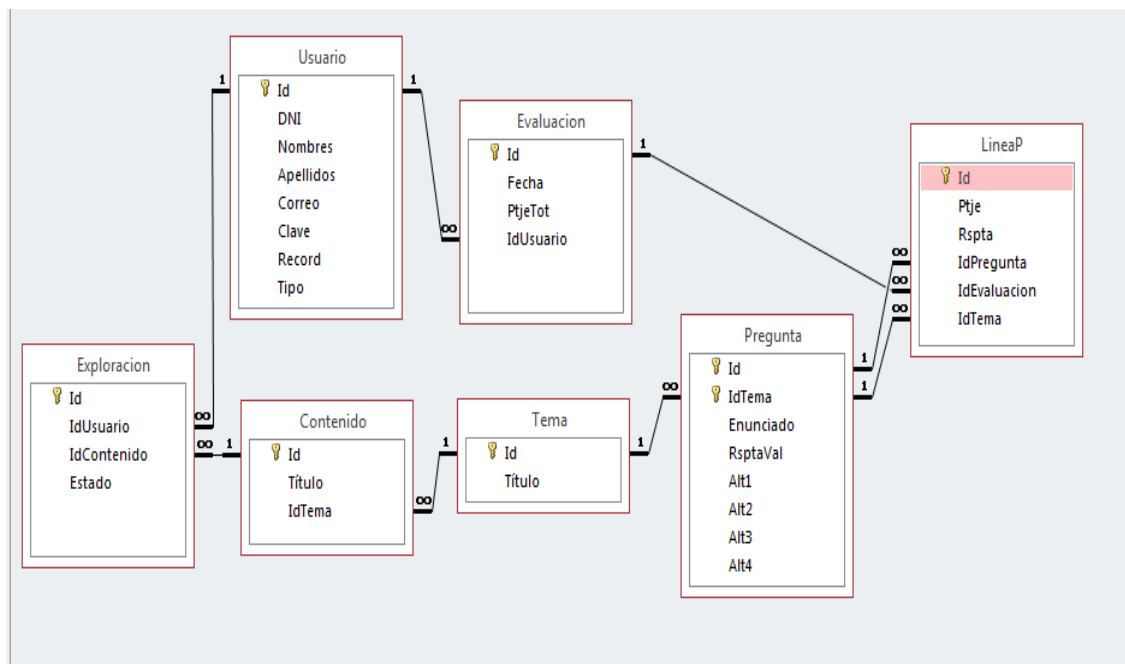


Ilustración 33: Base de datos del SAEMEF

En lo siguiente se procedió a generar las ventanas y el desarrollo mismo del código necesario para el MEC.

En las siguientes figuras, se puede observar las ventanas desarrolladas para el fin del material educativo computarizado:

- Ventana de ingreso a la aplicación

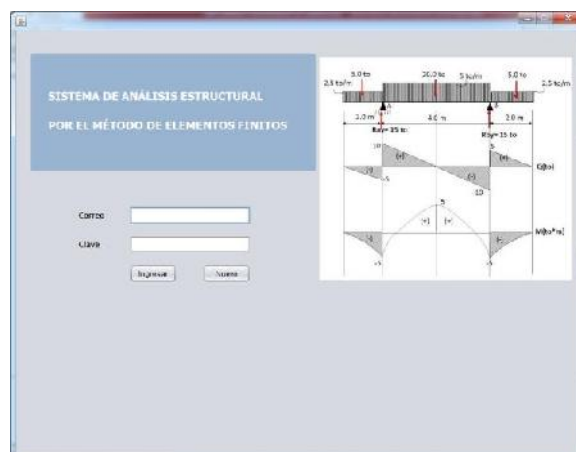


Ilustración 34: Ventana de Ingreso

- Ventana de registro de usuario

En la ventana de Inicio, ingresando los datos, por favor, ingresar son obligatorios. El usuario ya registrado solo podrá cambiar su clave.

Ilustración 35: Ventana de Registro de usuario

- Ventana de explorar el contenido

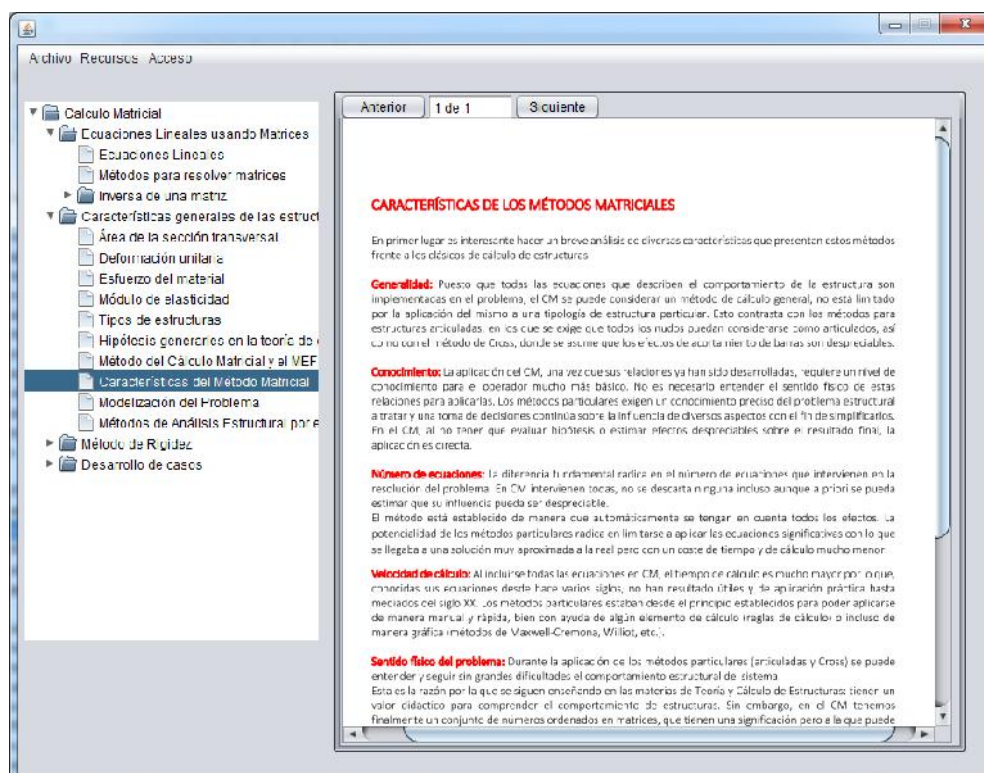


Ilustración 36: Ventana de revisar el contenido

- **Ventana de evaluación**

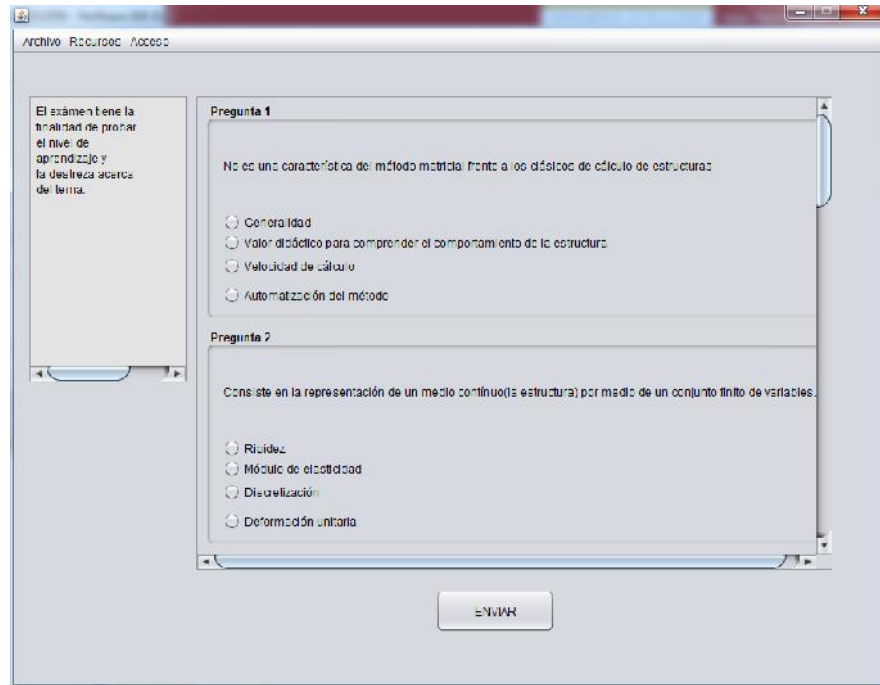
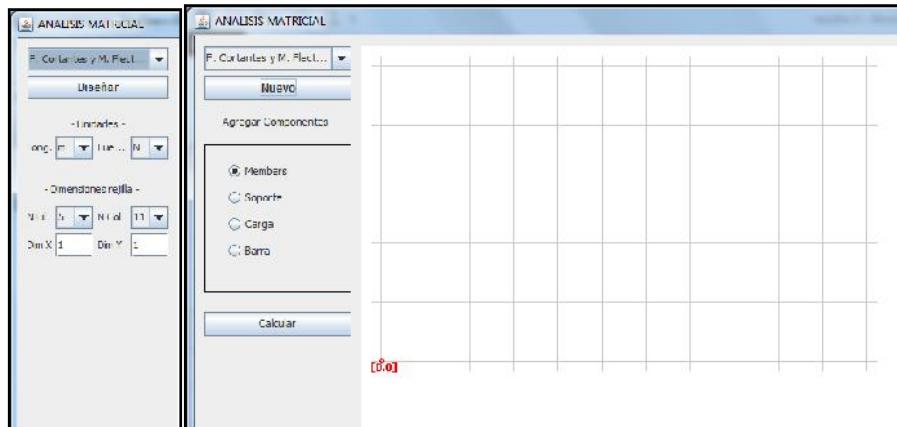


Ilustración 37: Ventana de evaluación

- **Simulador del análisis estructural.**

Durante este proceso se hizo comprobar el simulador del CalculoMatricial en Java que funcionara correctamente, por lo que se hizo en generar un caso en la que sus resultados se imprimieran tras generar el cálculo. De esta manera se pudo comprobar que los resultados eran los mismos en ejemplos planteados en libros y software de ingeniería. Para esto se consideró valores de entrada (Ilustración 38: A, B y C)

A) Definir rejilla B) Diseñar rejilla



C) Agregar componentes con valores según diseño de ejemplo.

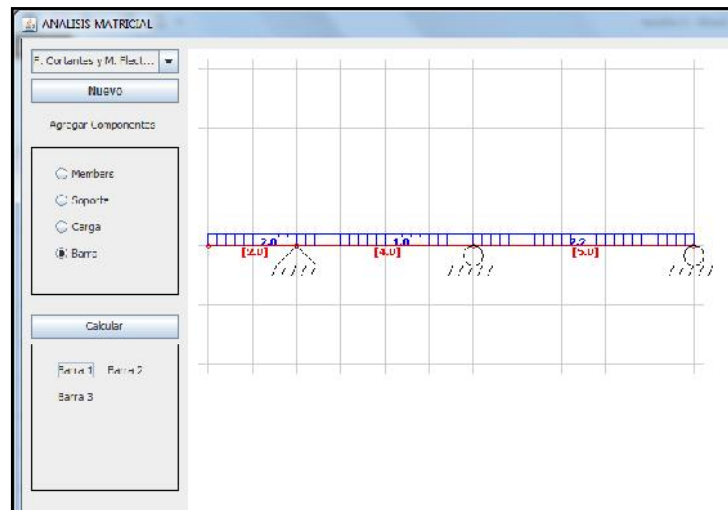


Ilustración 38: Pasos para el ingreso de datos en el simulador

En la ilustración 39, se muestra la salida de datos generado tras la ejecución de la clase `CalculoMatricial.java`

Reacciones y Momentos en cada Tramo
Tramo 1: R= 2.0 M= 0.667 R= 2.0 M= -0.667
Tramo 2: R= 2.0 M= 1.333 R= 2.0 M= -1.333
Tramo 3: R= 5.5 M= 4.583 R= 5.5 M= -4.583

Matriz Global

1.5	1.5	-1.5	1.5	0.0	0.0	0.0	0.0
1.5	2.0	-1.5	1.0	0.0	0.0	0.0	0.0
-1.5	-1.5	1.688	-1.125	-0.187	0.375	0.0	0.0
1.5	1.0	-1.125	3.0	-0.375	0.5	0.0	0.0
0.0	0.0	-0.187	-0.375	0.284	-0.135	-0.096	0.24
0.0	0.0	0.375	0.5	-0.135	1.8	-0.24	0.4
0.0	0.0	0.0	0.0	-0.096	-0.24	0.096	-0.24
0.0	0.0	0.0	0.0	0.24	0.4	-0.24	0.8

Vector Fuerza Externa
R= 2.0 M= 0.667 R= 4.0 M= 0.667 R= 7.5 M= 3.25 R= 5.5 M= -4.583

Matriz Reducida

1.5	1.5	0.0	0.0
1.5	2.0	0.0	0.0
1.5	1.0	0.5	0.0
0.0	0.0	0.5	1.8
0.0	0.0	0.4	0.8

Matriz Fuerza Reducida
2.0 0.667 0.667 3.25 -4.583

Matriz Desplazamiento Reducida
-14.426 7.88 5.213 -5.093 8.275

Matriz Desplazamiento Total
-14.426 7.88 0.0 5.213 0.0 -5.093 0.0 8.275

Matriz Fuerza Total
2.0 0.667 4.0 0.667 7.5 3.25 5.5 -4.583

Vector Fuerza y Momento - Tramo 1
0.0 0.0 4.0 -4.0

Vector Fuerza y Momento - Tramo 2
2.045 4.0 1.955 -3.819

Vector Fuerza y Momento - Tramo 3
6.264 3.819 4.736 0.0

Ilustración 39: Salida de datos de la aplicación

El prototipo del sistema se hizo en un principio en una hoja de cálculo para indicar a grandes rasgos las secciones que el sistema debía tener. La ilustración 40 (A Y B) muestra el esquema que se hizo.

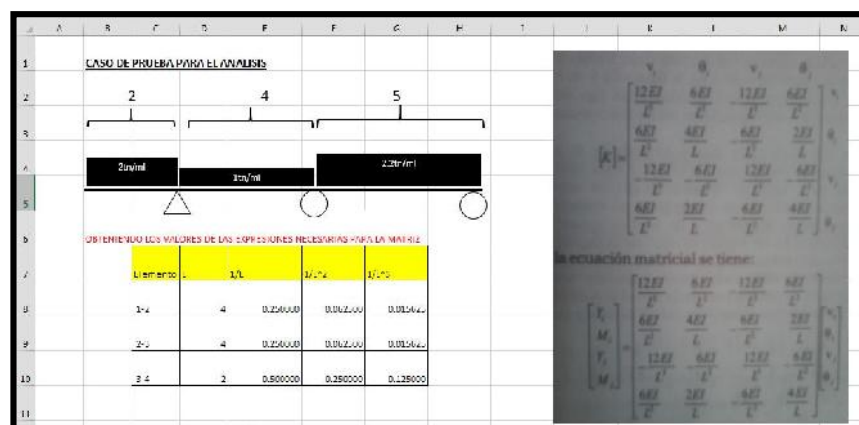


Ilustración 40.A) Diseño del esquema - prototipo

	B	C	D	E	F	G	H	I	J	K	L	M
13												
14	MATRICES LOCALES											
15												
16												
17												
18												
19												
20		U1	V1	U2	V2				U2	V2	U3	V3
21		u1	v1	u2	v2				u1	v1	u2	v2
22		0.18750	0.37500	-0.18750	0.37500	u1			0.1875	0.3750	-0.1875	0.3750
23		0.37500	1.00000	0.37500	0.50000	v1		k23	0.3750	1.0000	0.3750	0.5000
24		-0.18750	-0.37500	0.18750	-0.37500	u2		1	-0.1875	-0.3750	0.1875	-0.3750
25		0.37500	0.50000	-0.37500	1.00000	v2			0.3750	0.5000	-0.3750	1.0000
26												
27												
28												
29												
30												
31												
32												
33												
34	MATRIZ GLOBAL											
35												
36		U1		U1	V1	U2	V2	U3	V3	U4	V4	
37		V1		0.18750	0.37500	-0.1875	0.375	0	0	0	0	0
38		U2		0.37500	1.00000	-0.375	0.5	0	0	0	0	0
39		V2		-0.18750	-0.37500	0.37500	0.00000	-0.1875	0.3750	0	0	0
40		U3		0.37500	0.50000	0.00000	2.00000	0.3750	0.5000	0	0	0
41		V3		0	0	-0.1875	-0.3750	1.6875	1.1250	-1.5000	1.5000	
42		U4		0	0	0.3750	0.5000	1.1250	3.0000	-1.5000	1.0000	
43		V4		0	0	0	0	-1.5000	-1.5000	1.5000	-1.5000	
44				0	0	0	0	1.5000	1.0000	1.5000	2.0000	

Ilustración 40.B: Análisis del prototipo propuesto en Excel

Existen herramientas más especializadas para hacer este tipo de prototipos. Sin embargo, para el primer prototipo con esta hoja de cálculo fue suficiente, pues usar otras herramientas hubiera implicado más tiempo para tener el prototipo. Este enfoque sencillo permitió agilizar las cosas, sin tener un impacto en la calidad del sistema. Este prototipo ayudó a identificar las variables de entrada y de salida principalmente.

En la ilustración 41 se muestran los resultados implementados en excel, los mismos que se muestran en la parte inferior.

	Va	Θa	Vb	Θb		Des. Total	Fuerza	Fuerza -Momento	
u1	0.18750	0.37500	-0.18750	0.37500		-14.426	2	0	
v1	0.37500	1.00000	-0.37500	0.50000	x	7.88	+ 0.667	0	Tramo 1
u2	-0.18750	-0.37500	0.18750	-0.37500		0	2	4	
v2	0.37500	0.50000	-0.37500	1.00000		5.213	-0.667	-4	
	Vb	Θb	Vc	Θc					
U2	0.1875	0.3750	-0.1875	0.3750		0	2	2.05	
V2	0.3750	1.0000	-0.3750	0.5000		5.22	1.33	4	Tramo 2
U3	-0.1875	-0.3750	0.1875	-0.3750		0	2	1.95	
V3	0.3750	0.5000	-0.3750	1.0000		5.09	-1.33	-3.81	
	Vc	Θc	Vd	Θd					
u3	0.0960	0.2400	-0.0960	0.2400		0	5.5	6.26	
v3	0.2400	0.8000	-0.2400	0.4000		-5.09	4.58	3.81	Tramo 3
u4	-0.0960	-0.2400	0.0960	-0.2400		0	5.5	4.74	
v4	0.2400	0.4000	-0.2400	0.8000		8.27	-4.58	0	

Ilustración 41: Salida de datos del prototipo en Excel

Se calculó el porcentaje de error para todas las variables, considerando que el valor real era el del programa en Excel y el valor aproximado era el del programa en Java. El porcentaje de error es mínimo, pero no es 0. Se determinó que esta diferencia en los resultados se debe a Excel y Java usan diferente número de bytes para almacenar los valores de tipo real de doble precisión (double). Es por esto que, al ir realizando los cálculos, el resultado se desvía muy poco.

Una vez hecho la interfaz del material educativo computarizado, se comenzó a desarrollar contenido en cuatro capítulos correspondientes al aprendizaje del MEF para el análisis estructural.

1. Cálculo de Sistema de Ecuaciones Lineales usando matrices
 - Ecuaciones Lineales
 - Métodos para resolver matrices
 - Inversa de una Matriz
 - Determinante de una matriz.
 - Matriz de cofactores.
 - Matriz transpuesta.
2. Características generales de las estructuras
 - Área de la sección transversal
 - Deformación unitaria
 - Esfuerzo del material
 - Módulo de elasticidad
 - Tipos de estructuras
 - Hipótesis generales en la Teoría de Estructuras.
 - Método del Cálculo Matricial y el Método de Elementos Finitos
 - Características del método matricial
 - Modelización del problema
 - Métodos de análisis estructural por el cálculo matricial
3. Método de rigidez
 - Análisis estructural por el método general de rigidez.

- Condiciones de contorno
 - Elementos de sustentación: apoyos
 - Cargas nodales
 - Matriz de rigidez
 - Enfoque matricial
 - Características de la matriz de rigidez
 - Matriz de rigidez elementos tipo barra. Enrejados
 - Matriz de rigidez elementos tipo viga
4. Desarrollo de casos
- Ejemplo 1
 - Ejemplo 2

Ya con todo el contenido del curso almacenado en la base de datos, la evaluación (ejercicios) y de más actividades para el aprendizaje, se realizó la evaluación de los expertos de contenido y de informática para determinar la eficiencia del gestor de contenido SAEMEF.

El contenido del curso fue evaluado por el experto en contenido, velando así que el contenido del material sea relevante al área de conocimiento; es decir, que sea preciso, organizado y completo.

El especialista en informática por su parte, determinó un buen uso del potencial del computador, exponiendo además que dicho gestor de contenido era eficiente y eficaz en su desarrollo y en la documentación del sistema. Así como, todas las funciones previstas según el tipo de usuario, se desarrollaron eficientemente.

Al finalizar la evaluación de los expertos, se procedió hacer los ajustes necesarios para modificar lo que se encontraba con deficiencia y nuevamente se volvió a revisar.

Al tener el gestor de contenido cumpliendo las funciones programadas para cada uno de los usuarios, se procede a realizar los manuales de los distintos usuarios, en este caso, el Manual del Alumno y el Manual del Profesor.

Dichos manuales son sencillos, claro y completos, con los que se trata de solucionar los problemas a los que se puedan enfrentar durante la utilización del tutorial.

Antes de dar inicio a la prueba piloto, se hizo una revisión del tutorial con la técnica llamada Uno a Uno, para determinar problemas en la interfaz del MEC. Esta prueba fue realizada con un grupo de usuarios que cumplieran con características similares a la población objeto de estudio.

FASE VII: PRUEBA PILOTO DEL MATERIAL EDUCATIVO.

Con la prueba piloto se depuró el MEC. Dicha prueba fue aplicada a una muestra representativa de los diferentes tipos de usuario. Dicha muestra representativa fue seleccionada al azar.

A cada uno de los participantes de la prueba, se hizo entrega de la misma una vez que interactuaron con el MEC, dándole instrucciones de dejar en blanco aquello que no sepan la respuesta; con un tiempo ilimitado, de tal manera que la interacción sea a gusto de ellos mismos.

Una vez aplicada la prueba piloto, se procedió a realizar el análisis de los resultados, de donde se obtuvo información referente al tiempo de las sesiones, el tiempo en las evaluaciones del MEC y el tiempo para el cumplimiento total del curso.

Finalmente se resolvieron las dificultades y exigencias reflejadas por los usuarios durante la prueba piloto. Deduciendo que el MEC está disponible para ser usado por la población para la que fue desarrollado.

RESULTADOS Y DISCUSIÓN

Los cuestionarios aplicados a usuarios representativos y expertos se diseñaron de acuerdo a los formatos propuestos por Galvis (1992) con rangos de evaluación correspondientes a la escala de Likert:

(3) Totalmente de acuerdo (2) Medianamente de acuerdo (1) Total desacuerdo

Resultados de la prueba aplicada a expertos.

Se realizaron dos pruebas con usuarios expertos de la aplicación. La primera de ellas con un experto en contenido, como lo es director de las carreras de ingenierías..

Esta prueba constó de ocho (8) preguntas relativas a la disposición de los contenidos de los temas.

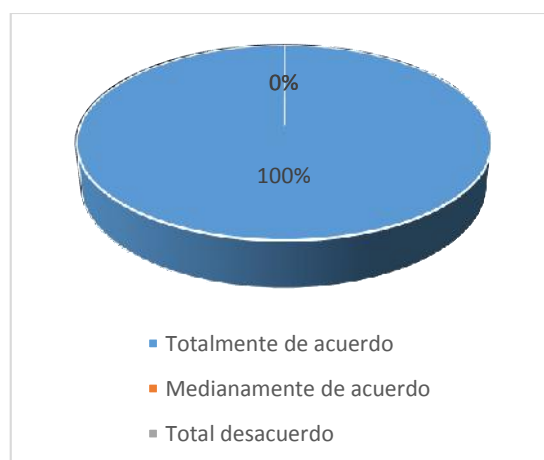
Para el ingeniero consultado, los contenidos fueron suficientes para estudiar el análisis estructural por el método de elementos finitos, adquiriendo un excelente grado de claridad y precisión; siendo presentados de manera concisa y organizada. Estos resultados se muestran en la Tabla 7.

Tabla 7. Resultados de la prueba con usuarios expertos en contenido.

N°	Preguntas	3	2	1
1	Los contenidos de la aplicación son suficientes para entender el análisis estructural por el MEF.	1	0	0
2	La aplicación hace que sus contenidos adquieran un excelente grado de claridad y precisión.	1	0	0
3	La información que presenta la aplicación sobre el análisis estructural por el MEF es clara y concisa.	1	0	0
4	El contenido de la aplicación está actualizado.	1	0	0
5	El contenido de la asignatura en la aplicación está lógicamente organizado.	1	0	0
6	Los contenidos son suficientes para lograr los objetivos si el usuario tiene los conocimientos previos.	1	0	0
7	Los temas son relevantes como apoyo al contenido.	1	0	0
8	El contenido de la aplicación es transferible a otras asignaturas.	1	0	0
	Total	8	0	0

Los datos arrojados por este instrumento fueron ponderados y se presentan en la Ilustración 42.

Ilustración 42. Resultados de la prueba con usuarios expertos en contenido.



Según el gráfico anterior se tiene que, la experta consultada en contenido se encuentra en total acuerdo con todos los criterios evaluados de la aplicación educativa multimedia.

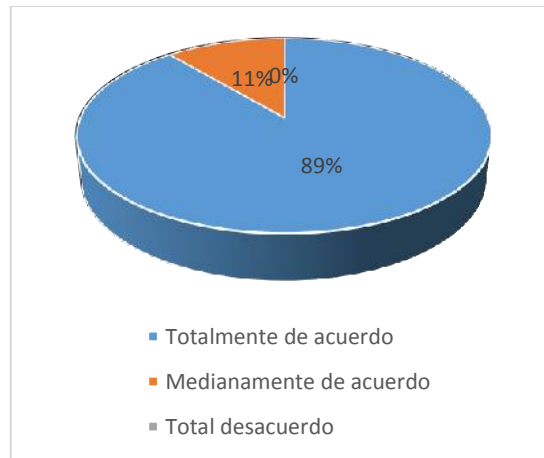
La segunda prueba se efectuó con un especialista en Informática.

Tabla 8. Resultados de la prueba con usuarios expertos en metodología.

N°	Preguntas	3	2	1
1	La aplicación está desarrollada de forma apropiada para la audiencia a quien se dirige.	1	0	0
2	El usuario de la aplicación siempre sabe dónde está dentro del desarrollo del contenido.	1	0	0
3	El tipo de preguntas que hace la aplicación en las autoevaluaciones es adecuado.	1	0	0
4	El nivel de exigencia en las autoevaluaciones corresponde al contenido mostrado en la aplicación.	1	0	0
5	Las explicaciones dadas en la autoevaluación son claras.	1	0	0
6	Los colores usados en la aplicación son adecuados.	1	0	0
7	El tipo de letra y su color permiten leer con facilidad el contenido de la aplicación.	0	1	0
8	Los gráficos y efectos visuales ayudan a entender el contenido de la aplicación.	1	0	0
9	La aplicación es sencilla de usar.	1	0	0
	Total	8	1	0

Los datos arrojados por este instrumento se ponderaron según la Ilustración 43.

Ilustración 43. Resultados de la prueba con usuarios expertos en metodología.



El gráfico anterior muestra como los expertos en metodología se encuentran en total acuerdo con todos los criterios evaluados de la aplicación educativa multimedia, a excepción de la pregunta 7, donde se obtuvo un voto para medianamente de acuerdo.

En las observaciones verbales y sugerencias de los profesionales, se indicó que los aspectos con los que estuvieron medianamente de acuerdo pueden cambiar a totalmente de acuerdo en tanto se corrijan las observaciones concretas realizadas.

Resultados de la prueba aplicada a los usuarios representativos

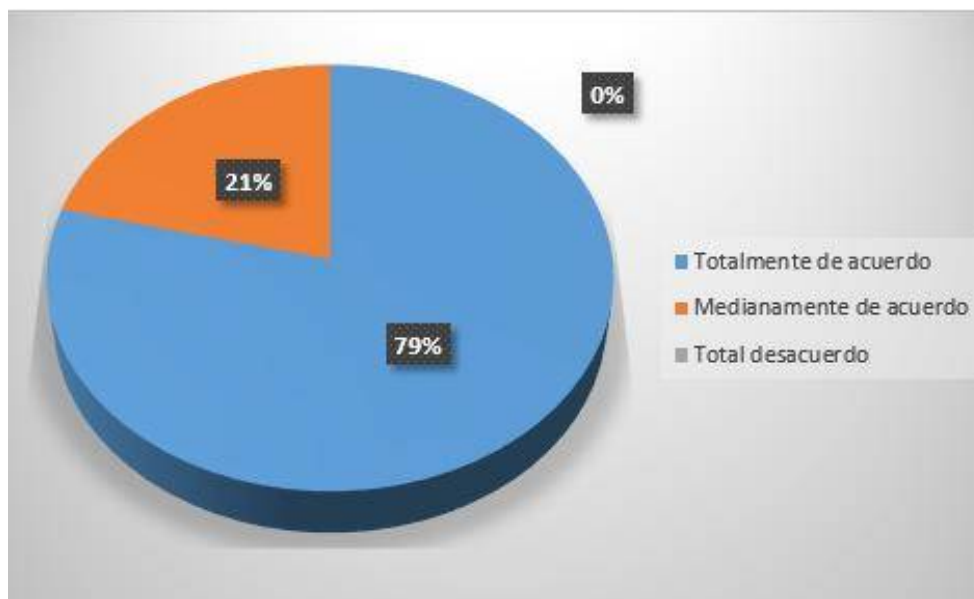
La prueba con los usuarios representativos de la aplicación constó de trece (13) preguntas y los resultados obtenidos se muestran en la Tabla 9.

Tabla 9. Resultados de la prueba con usuarios representativos.

N°	Preguntas	3	2	1
1	La aplicación le permite a usted avanzar a su propio ritmo de aprendizaje	11	1	0
2	El uso de la aplicación es estimulante.	9	3	0
3	Los nombres que identifican los botones corresponden con su contenido.	12	0	0
4	Los colores usados en la aplicación son adecuados.	12	0	0
5	El tipo de letra y su color permiten leer con facilidad el contenido de la aplicación.	7	5	0
6	La cantidad de información por pantalla es adecuada.	7	5	0
7	El uso de gráficos y efectos visuales permiten entender el contenido.	9	3	0
8	El tipo de preguntas que se hace en la autoevaluación es adecuado.	7	5	0
9	El nivel de exigencia en las autoevaluaciones corresponde al contenido mostrado en la aplicación.	11	1	0
10	La información de retorno, dada en la autoevaluación, es suficiente para saber cuánto estaba aprendiendo.	7	5	0
11	Las explicaciones dadas en la autoevaluación son “amigables”.	7	5	0
12	La presentación de la aplicación es “amigable”.	12	0	0
13	La aplicación es sencilla de usar.	12	0	0
	Total	117	39	0

Los datos arrojados por este instrumento fueron ponderados de manera gráfica en la Ilustración 44 y posteriormente analizados con el fin de verificar el puntaje porcentual de aprobación que tuvo la aplicación educativa para los usuarios representativos.

Ilustración 44. Resultados de la prueba con usuarios representativos.



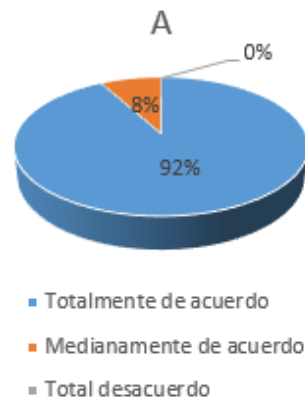
Como se refleja en la Ilustración 44, el 79% de los encuestados manifestó estar totalmente de acuerdo con la aplicación educativa multimedia desarrollada y siguiendo los resultados de la Tabla 9, los criterios de las preguntas 1, 3, 9, 12 y 13 fueron los mejores valorados. Por su parte, el otro 21% está medianamente de acuerdo con esos criterios. No se obtuvo resultados en total desacuerdo con los enunciados evaluados.

De manera complementaria al análisis anterior, se analizaron los resultados de la prueba en base a 5 criterios: navegación, motivación, interfaz, contenido y cuestionarios.

Las preguntas 1 y 13 están relacionadas con el criterio de navegación. La Ilustración 45. (a) muestra como el 92% de los consultados está totalmente de acuerdo con que la aplicación le permite avanzar a su propio ritmo; mientras que el restante 8% está medianamente de acuerdo con ello. Por su parte, según la misma figura, en la sección (b), se muestra como se obtuvo el 100% en total acuerdo con que la aplicación es sencilla de usar.

Ilustración 45. Evaluación de la navegación, según usuarios representativos.

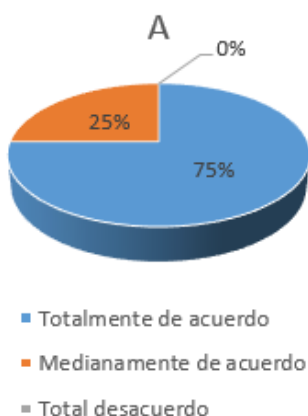
a) La aplicación le permite a usted avanzar a su propio ritmo de aprendizaje, b) La aplicación es sencilla de usar.



El criterio de motivación se evaluó con las preguntas 2 y 12. La Ilustración 46, en su sección (a), muestra como las dos terceras partes de los consultados (75%) están totalmente de acuerdo con que la aplicación es estimulante (pregunta 2) y la otra tercera parte (25%) está medianamente de acuerdo. Los resultados de la misma figura en su sección (b), dicen que el 100% está totalmente de acuerdo con que la aplicación es agradable (pregunta 12). Considerando que la aplicación se considera agradable y estimulante, entonces podría resultar motivadora.

Ilustración 46. Evaluación de la motivación, según usuarios representativos.

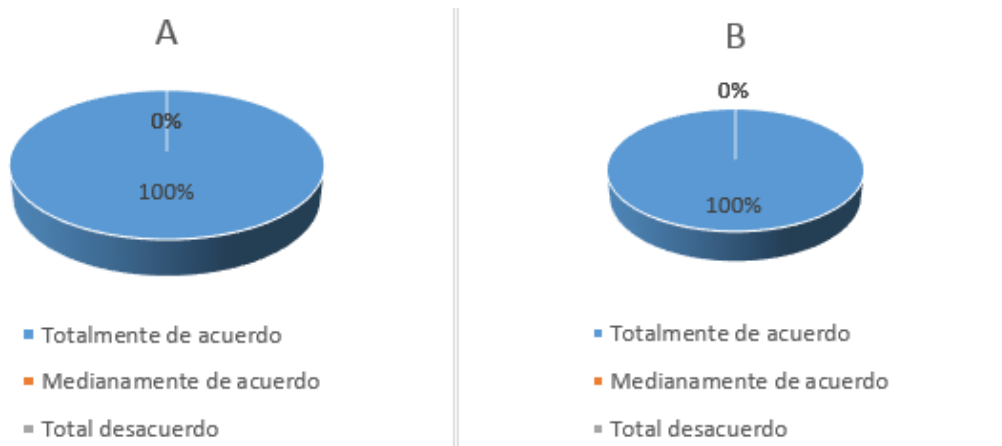
a) El uso de la aplicación es estimulante, b) La aplicación es agradable.



La interfaz de la aplicación se evaluó con las preguntas 3 y 4. La pregunta 3, cuyos resultados se muestran en la Ilustración 47(a), obtuvo 100% de total aceptación; mientras que en la Ilustración 47(b) se muestra como en la pregunta 4 también estuvieron en total acuerdo.

Ilustración 47. Evaluación de la interfaz, según usuarios representativos.

- a) Los nombres que identifican los botones corresponden con su contenido, b) Los colores usados en la aplicación son adecuados.



Según las apreciaciones verbales y directas emitidas luego de efectuar la prueba, los consultados expusieron que, en general, la combinación de colores estuvo acertada

Por otra parte, se exaltó el uso correcto de los colores en los botones y en el texto para distinguir aspectos importantes del contenido cuando se requería. Los consultados consideraron que esto permitía resaltar e identificar de manera asertiva elementos especiales de dicho contenido.

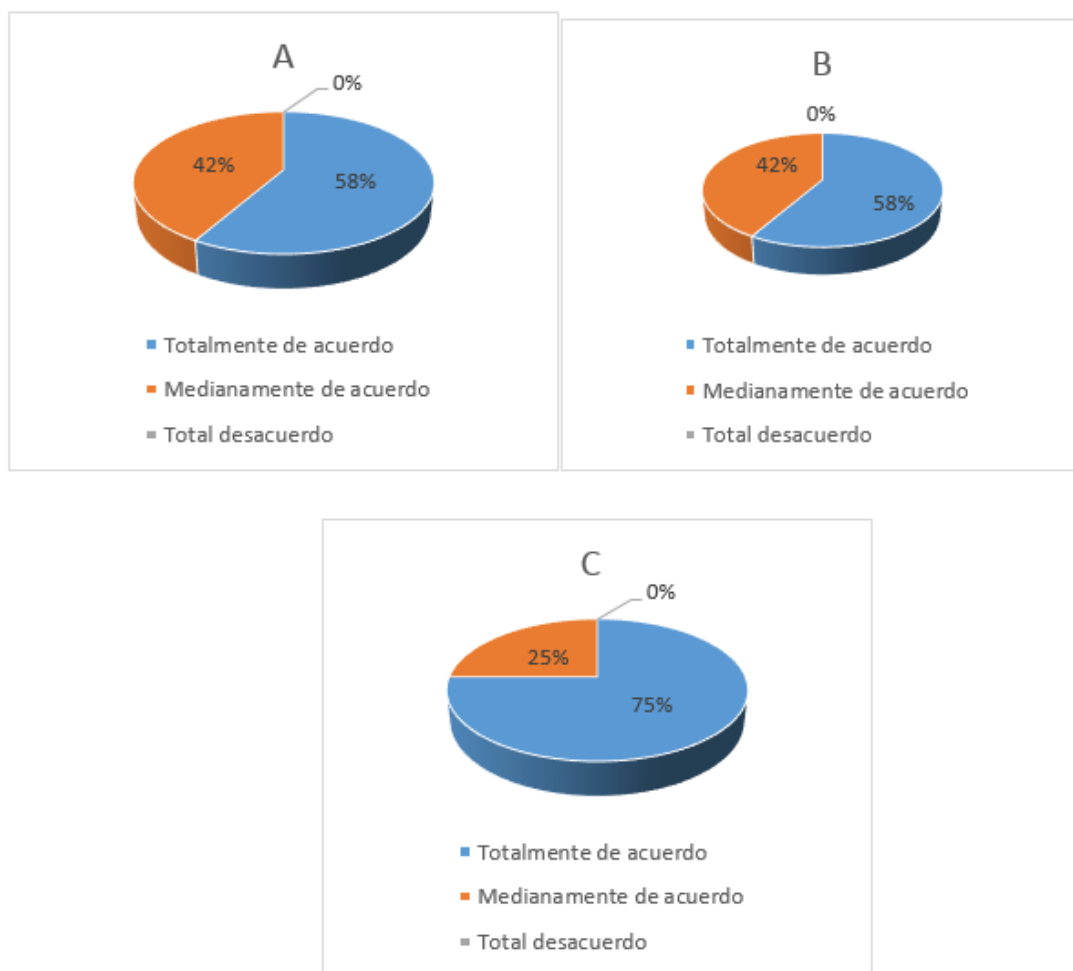
El cuarto criterio de evaluación se refiere al contenido y se evaluó con las preguntas 5, 6 y 7. Los resultados de cada una de ellas se aprecian en la Ilustración 48, en sus secciones (a), (b) y (c), respectivamente. En las dos primeras de ellas se obtuvo una valoración de 58% de total acuerdo con su respectivo criterio y 42% con medianamente de acuerdo. La pregunta 7 obtuvo un 25% de mediana aceptación y un 75% de total aceptación.

El último criterio evaluado fue cuestionario y para ello se utilizaron las preguntas 8, 9, 10 y 11, relativas a este tema. Los consultados dijeron estar en total acuerdo, en un 58%, con que: el tipo de preguntas de las autoevaluaciones es adecuado, la información de retorno es suficiente para saber cuánto se está aprendiendo y que las explicaciones dadas son claras. Por su parte, el restante 42%, está medianamente de acuerdo con estos criterios. Referente a la pregunta 8, sobre el nivel de exigencia de los cuestionarios, el 92% estuvo

totalmente de acuerdo con que éste corresponde con el contenido mostrado en la aplicación y sólo el 8% estuvo medianamente de acuerdo.

Ilustración 48. Evaluación del contenido, según usuarios representativos.

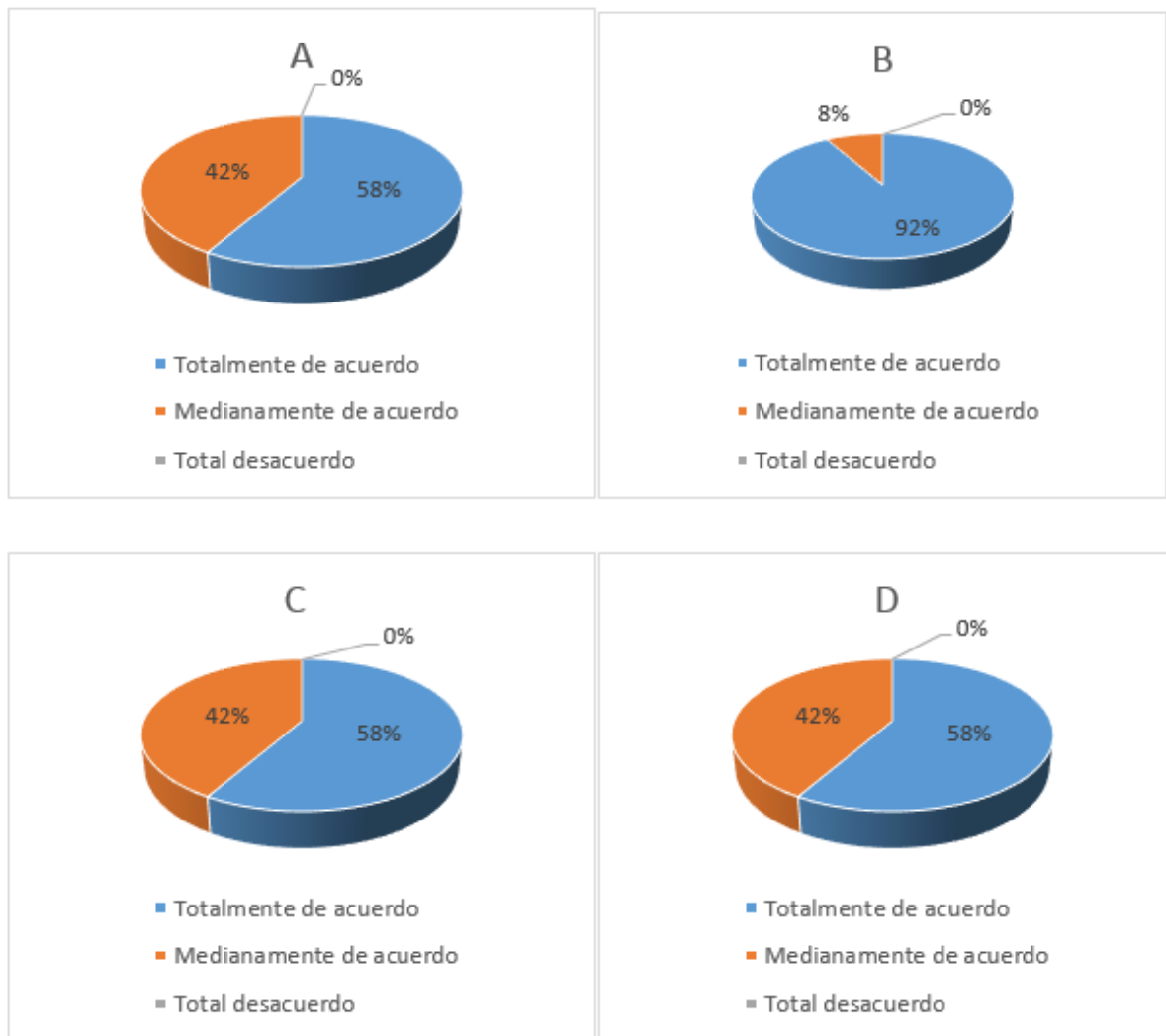
- a) El tipo de letra y su color permiten leer con facilidad el contenido de la aplicación,
- b) La cantidad de información por pantalla es adecuada,
- c) El uso de gráfico y efectos visuales permiten entender el contenido.



Los resultados del criterio cuestionario se reflejan en la Ilustración 49, en sus secciones (a), (b), (c) y (d), correspondientes con los resultados de las preguntas 8, 9, 10 y 11, respectivamente.

Ilustración 49. Evaluación de los cuestionarios, según usuarios representativos.

- a) El tipo de preguntas que se hace en las autoevaluaciones es adecuado,
- b) El nivel de exigencia en las autoevaluaciones corresponde al contenido mostrado en la aplicación,
- c) La información de retorno, dada en la autoevaluación, es suficiente para saber cuánto se estaba aprendiendo,
- d) Las explicaciones dadas en la autoevaluación son claras.



Cabe destacar que, para ninguno de los criterios evaluados en la prueba efectuada con usuarios representativos de la aplicación, se obtuvo resultados en total desacuerdo, y con un 71% en total acuerdo, se considera aprobado el software desde el punto de vista de los usuarios.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Habiendo profundizado en el ejercicio del análisis estructural para lograr desarrollar un contenido pertinente para el aprendizaje del mismo y con el desarrollo de un simulador como complemento y apoyo se pudo definir los mecanismos y características principales del método de elementos finitos; mecanismos predecibles y sistemáticos que desde luego se con apoyo de un computador ante estructuras complejas nos puede brindar datos lo más cercano al comportamiento de cualquier estructura.
- A través metodología de Álvaro Galvis se conservan los grandes pasos o etapas de un proceso sistemático para desarrollo de software educativo (análisis, diseño, desarrollo, prueba y ajuste, implementación). Sin embargo, se da particular énfasis a los siguientes aspectos: la solidez del análisis, como punto de partida; el dominio de teorías sustantivas sobre el aprendizaje y la comunicación humanas, como fundamento para el diseño de los ambientes educativos computarizados; la evaluación permanente y bajo criterios predefinidos, a lo largo de todas las etapas del proceso, como medio de perfeccionamiento continuo del material; la documentación adecuada y suficiente de lo que se realiza en cada etapa, como base para el mantenimiento que requerirá el material a lo largo de su vida útil.
- En el campo de la informática, surgen variadas herramientas cada día con componentes y métodos innovadores, es por ello que es importante saber que herramientas de desarrollo hay en el momento de realizar el proyecto, pues de esta manera se pueden ofrecer mejores servicios y se aprovechar al máximo las arquitecturas y demás estrategias implantadas. Java como plataforma para el desarrollo de aplicaciones motivadoras en la enseñanza y aprendizaje, es eficiente, por cuanto permite el desarrollo de entornos dinámicos por medio del uso de sus distintos componentes que incorpora.
- Al realizaron varias pruebas se pudo determinar el desempeño óptimo del tutorial y el cumplimiento del objetivo para lo cual fue el desarrollo del mismo. Los estudiantes que aprueben el tutorial, tendrán una base para analizar estructuras de mayor dimensión contando con criterios ingenieriles necesarios de quienes pretenden especializarse y establecerse como buenos profesionales de la ingeniería.

RECOMENDACIONES

- En atención a lo expuesto anteriormente, se evidencia en afirmar que la educación unida a la informática es un binomio que apoya en el aprendizaje de los contenidos de cualquier asignatura, como es el caso del análisis estructural por el método de elementos finitos. Sin embargo, la presencia de software educativo en el campo de la ingeniería por sí solo no supone un salto cualitativo en el ámbito de la educación. Pensar de esta manera es una desvalorización de los docentes y su interrelación con los alumnos. Por sí mismos no han de modificar ni los contenidos, ni los métodos, ni la calidad de los aprendizajes. En realidad, su uso en educación introduce como una herramienta más, que debe ser incorporada de acuerdo al tema y los objetivos buscados por la enseñanza de manera complementaria con otras herramientas pedagógicas.
- Se recomienda no dejar morir el proyecto, se puede ir incorporando niveles al estudio para mejorar el aprendizaje de esta metodología para casos de mayor envergadura.
- Así como el objetivo de este tutorial era difundir el método de elementos finitos para el método de la rigidez en el análisis estructural, también se pueden incorporar a otras materias, otros temas y/u otras áreas educativas.

REFERENCIAS BIBLIOGRÁFICAS

- Pezzotti , S., & Antico, F. (2007). *Introducción a la Teoría de Elementos Finitos*. La Plata Argentina.
- A, G. (1992). *Ingeniería del Software Educativo*. Santa Fe de Bogotá. Colombia: Ediciones.
- Arias, F. (2006). *Introducción a la Metodología Científica*. Episteme. 5ta Edición.
- Bertoglio, J. O. (2004). *Introducción a la teoría general de sistemas*. Limusa.
- Celigüeta Lizarza, J. T. (Setiembre de 2008). *Método de los Elementos Finitos para Análisis Estructural*. Gipuzkoa, San Sebastián, España.
- Delgado Contreras , G. (s.f.). *Diseño de Estructuras Aporticadas de Concreto Armado*. Lambayeque: EdiCivil SRL.
- Kendall, K. K. (s.f.). *Análisis y Diseño de Sistemas de Información*. Editorial Prentice Hall. Tercera Edición.
- Miller, J. (s.f.). *The Living Systems*.
- Paico Gasco, S. A. (2013). *Método Matricial en el Analisis Estructural*. Chiclayo: JASV Producciones S.R.L.
- Peña, L. M. (2002). *Tutorial Interactivo con Tecnología Multimedia para el Aprendizaje del Lenguaje Java*. Maracay.
- Salcedo, P. (26 de abril de 2010). Ingeniería de software educativo, teorías y metodologías que la sustentan. *Revista de Ingeniería Informática*.