

MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

NEW OBJECT-ORIENTED MODEL FOR STRUCTURAL ANALYSIS OF PLANE FRAMES BY FINITE ELEMENTS

Roger Gustavo Saravia

Universidad Mayor de San Andres - Bolivia

rodgergustavus@gmail.com

Abstract: This research applies object-oriented (OO) analysis and design to structural engineering; specifically, analysis of rigid plane frames using the finite element method (FEM). Among the useful results of the structural analysis, we have the internal forces of the elements that are necessary to verify and improve the design of a given structure. Due to simplification reasons in order to save memory and time machine, the commercial software of structural analysis has been producing the results of the analysis using a single local coordinates system for both ends of the elements of a structure allowing to the engineer's interpretation the values for the other end of the element; which leads to ambiguities and errors. This article proposes the implementation of a program for structural analysis capable to produce results of the internal forces for each end of the elements of a structure. This implementation is based on a new OO model for the structural analysis using FEM - beam element with rigid joints and Kardestuncer local coordinates system.

Keywords: Abstraction; Agile; AlgorithmAnalysis; Asd; Axes; Beam; Cascade; Class.

Resumen: Ésta investigación se extiende en el área del análisis y diseño orientado a objetos (OO) de las ciencias de computación con campo de aplicación en la ingeniería estructural; específicamente, en el análisis de marcos rígidos planos mediante el método de los elementos finitos (MEF). Entre los resultados útiles del análisis de una estructura, están las fuerzas internas de los elementos que sirven para verificar y mejorar el diseño de la estructura sea de madera, hormigón armado, y otros. Por razones de simplificación que se daban antiguamente para ahorrar memoria y tiempo de máquina, el software comercial de análisis estructural ha venido produciendo los resultados del análisis referidos a un solo sistema de ejes de coordenadas locales para ambos extremos de los elementos de una estructura dejando a la interpretación del ingeniero los valores para el otro extremo del elemento lo cual conlleva a ambigüedades y errores. Esta investigación propone la implementación de un programa de cálculo estructural que produce resultados de las fuerzas internas para cada extremo de los elementos de una estructura. Dicha implementación se basa en un nuevo modelo OO para el análisis mediante el MEF – elemento viga doblemente empotrada sistema de ejes locales Kardestuncer.

Palabras Clave: Abstracción; Agiles; Algoritmo; Análisis; Apoyos;Asd; Cargas.

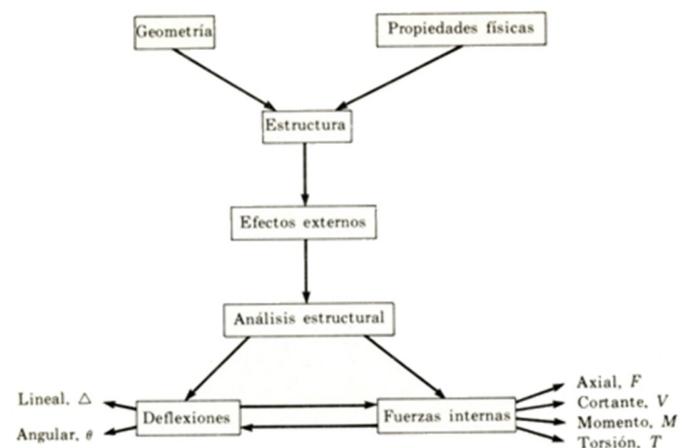
I INTRODUCCIÓN

Las estructuras materializadas en obras civiles, herramientas mecánicas, medios de transporte terrestres, aeroespaciales, navales, y otros. juegan un rol de servicio a diario muy

importante para la humanidad. Las estructuras se presentan en todo el Mundo a diversas escalas y complejidades de diseño como ser: cimentaciones, edificaciones, puentes, muros, carreteras, torres de transmisión, presas, canales, vehículos, aviones, barcos, máquinas, reactores nucleares, y otros

El análisis estructural es una rama de las ciencias físicas que trata sobre el comportamiento de las estructuras bajo ciertas condiciones de diseño. Las estructuras son sistemas que soportan carga y el comportamiento se entiende como su tendencia a deformarse, vibrar, pandearse o fluir dependiendo a las condiciones a las cuales estén sometidas. Los resultados del análisis se usan para determinar la deformación de la estructura y verificar si son adecuadas para las cargas para las cuales se las ha diseñado. El resultado es único y depende solamente de las condiciones iniciales. Véase figuras 1.

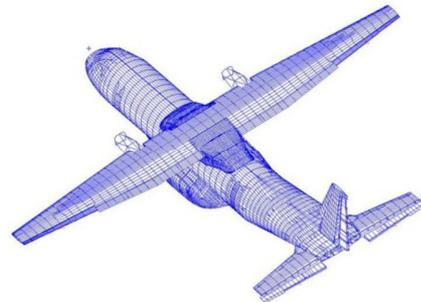
FIGURA 1. EL FIN ES DISEÑAR UNA ESTRUCTURA Y EL ANÁLISIS ESTRUCTURAL ES UNA DE LAS HERRAMIENTAS PARA ALCANZAR TAL FIN.



A. Método de los Elementos Finitos

El método de los elementos finitos (MEF) es la manera más extendida para simular sistemas de ingeniería en computadoras. El MEF ha sido usado para predecir el comportamiento de sistemas estructurales, mecánicos, térmicos, de flujo de fluidos, eléctricos y químicos. El MEF ha sido aplicado tanto a sistemas lineales o no lineales sean o no tiempo dependientes. El MEF se aplica a sistemas formados por cualquier material y geometría. En el Mundo se emplea anualmente más de un billón de dólares en gastos concernientes al uso del MEF. El MEF ha sido desarrollado por Jon Turner, ingeniero líder de la compañía BOEING, entre 1956 y 1962 con el fin de lograr un mejor cálculo para la rigidez de las alas de un avión destinado a vuelo supersónico.

FIGURA 2. MODELO DE ELEMENTOS FINITOS DE UN AVIÓN.



La tecnología MEF consiste básicamente en: una librería de elementos modelo básicos, un proceso para combinar todos estos modelos en un modelo matemático global del sistema de ingeniería en cuestión y un algoritmo para la resolución numérica de las ecuaciones del sistema.

Esta investigación está orientada a sistemas estructurales planos que soportan cargas en el mismo plano de los elementos (figura 3). Se usará el elemento finito viga doblemente empotrada con sistema de

En la figura 4 se tiene un elemento finito viga doblemente empotrada. Los modelos existentes, por cuestiones de simplificación, siempre han empleado un solo sistema de ejes locales en ambos extremos del elemento (figura 4a). Entonces, cuando se llega a los resultados del análisis, hay que hacer una serie de interpretaciones para llegar a las fuerzas internas reales porque el uso de un mismo sistema de ejes para ambos extremos genera ambigüedades como ser: ¿en qué extremo el eje local x indica la dirección de la tensión? algunos suponen que va de izquierda a derecha pero ¿cuál es el extremo izquierdo? Para evitar cualquier ambigüedad, en esta investigación se propone un sistema de ejes locales para cada extremo del elemento según la teoría estructural de Kardestuncer (figura 4b).

B. Método de la Rigidez

El método matricial de la rigidez es un proceso universalmente aceptado para el análisis estructural en computadoras que proporciona un algoritmo rápido y preciso para el análisis de estructuras complejas. El objetivo del método de la rigidez es establecer la relación entre cargas externas y desplazamientos en los nudos de la estructura. El método de la rigidez sustituye la estructura continua por un modelo equivalente compuesto por elementos discretos (i.e. finitos) cuyas propiedades elásticas e inerciales están expresadas matricialmente. Estas matrices conocidas como matrices de rigidez de los elementos constituyen los bloques constructivos básicos que al ser combinados en conjunto de acuerdo a una serie de reglas derivadas de la teoría de la elasticidad conforman la matriz de rigidez global que representa a la estructura.

ejes locales Kardestuncer. Y se aplicará el método matricial de la rigidez como proceso para obtener el modelo matemático global de la estructura

En la figura 4 se tiene un elemento finito viga doblemente empotrada. Los modelos existentes, por cuestiones de simplificación, siempre han empleado un mismo sistema de ejes locales en ambos extremos del elemento (figura 4a).

FIGURA 3. SISTEMA ESTRUCTURAL PLANO: UN PÓRICO O MARCO RÍGIDO DE 20 NUDOS, 28 ELEMENTOS VIGA Y 4 APOYOS FIJOS.

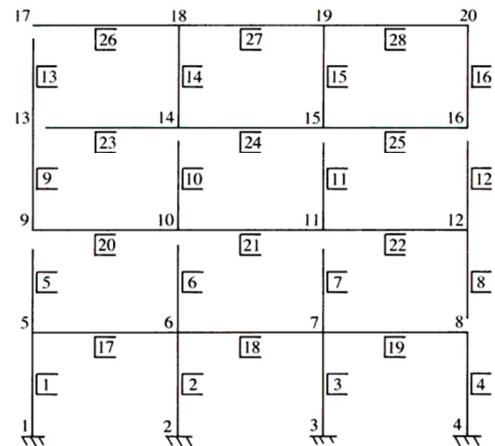
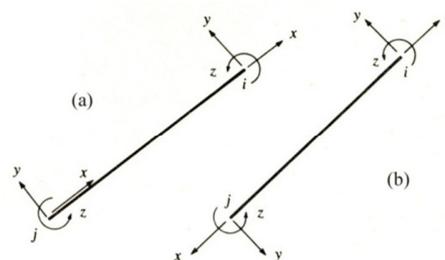


FIGURA 4. SISTEMAS DE EJES LOCALES: (A) TRADICIONAL (B) SEGÚN KARDESTUNCER.



MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

Los principios fundamentales sobre los cuales se sustenta el método de la rigidez son: los principios de desplazamientos y fuerzas virtuales, los teoremas de Castigliano, el teorema de energía mínima, el teorema de la energía mínima

complementaria y los teoremas del desplazamiento y carga unitarios(as).

FIGURA 5. MARCO RÍGIDO PLANO

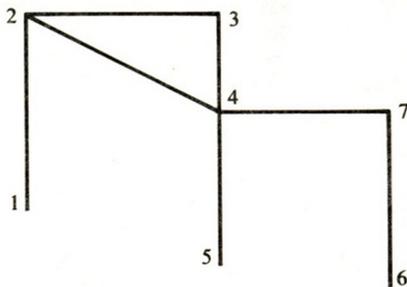


FIGURA 6. MODELO DE LA MATRIZ DE RIGIDEZ GLOBAL.

$$K = \begin{bmatrix} K_{11} & K_{12} & 0 & 0 & 0 & 0 & 0 \\ K_{21} & K_{22} & K_{23} & K_{24} & 0 & 0 & 0 \\ 0 & K_{32} & K_{33} & K_{34} & 0 & 0 & 0 \\ 0 & K_{42} & K_{43} & K_{44} & K_{45} & 0 & K_{47} \\ 0 & 0 & 0 & K_{54} & K_{55} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_{66} & K_{67} \\ 0 & 0 & 0 & K_{74} & 0 & K_{76} & K_{77} \end{bmatrix}$$

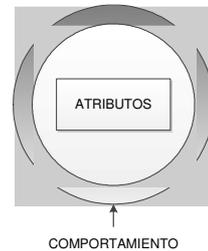
El ensamblado de la matriz de rigidez global del marco plano de la figura 5 está compuesto a su vez por las matrices de rigidez de cada elemento de la estructura. Cada componente K_{ij} de la matriz global K es una matriz 3×3 correspondiente al elemento que está entre los nudos i y j . En consecuencia la matriz de rigidez global para esta estructura es de orden 21. Cada sub-matriz K_{ii} de la diagonal principal representa la rigidez del nudo i o sea la

sumatoria de las matrices de rigideces de los elementos que parten del nudo i . La matriz global K siempre es simétrica. Si no existen elementos entre los nudos i y j entonces K_{ij} es cero

C. Enfoque Orientado a Objetos

Un objeto se define en dos términos: atributos y comportamiento. Un objeto es una entidad que contiene datos y comportamiento. Los objetos son los bloques constructivos de un programa orientado a objetos (OO). Un software que usa el paradigma OO consiste en una colección de objetos. En la programación procedimental tanto el código como los datos están localizados de manera separada. En un diseño OO, los atributos y el comportamiento están contenidos en un solo objeto. Por consiguiente, como en la programación procedimental los datos están separados de los procedimientos, muchas veces los datos están a nivel de contexto global de tal manera que pueden ser modificados por cualquier subrutina o función implicando que el acceso a los datos pueda ser incontrolado e impredecible. Entonces, como no se tiene el control de que subrutinas podrían haber accedido a los datos, las pruebas y depuración del software son muy difíciles.

FIGURA 7. ENCAPSULACIÓN EN UN OBJETO



Los objetos resuelven todos estos problemas al combinar datos y comportamiento en un paquete completo. La combinación de datos y comportamiento (métodos) en una misma entidad se denomina encapsulación (figura 7). Otros conceptos adicionales del paradigma OO son: encapsulación, ocultamiento, interfaz, implementación, herencia, superclases, subclases, abstracción, polimorfismo, composición, persistencia, clases abstractas, interfaces java, y otros.

debido a que su estructura es inherentemente poco acoplada. Esto lleva a menores efectos colaterales cuando se deben hacer cambios. Los sistemas OO son fáciles de adaptar y escalables (pueden crearse grandes sistemas ensamblando subsistemas reutilizables).

D. Problema Identificado

Como el análisis y diseño OO permite la creación y utilización de nuevas estructuras de datos, ha sido exitosamente empleado en todo tipo de aplicaciones. Las tecnologías OO llevan a reutilizar componentes de software lo que conlleva a un desarrollo de software rápido y de mejor calidad. El software OO es fácil de mantener

Programar el análisis estructural mediante el MEF en lenguajes basados en procedimientos implica una gran extensión de código distribuido en subrutinas y funciones. El análisis estructural involucra operaciones matriciales que incluyen a las propiedades del elemento y éstas deben implementarse mediante ciclos explícitos.

MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

En ingeniería del software se denomina “complejidad ciclomática” al grado de profundidad de las anidaciones. Operaciones como las asignaciones vectoriales deben programarse dimensión a dimensión generando ineficiencia. Debe reservarse considerable cantidad de variables para almacenar la configuración geométrica, las propiedades de los materiales y las cargas aplicadas en la estructura; esto, debido a una falta de tipos de datos específicos que permitan un alto nivel de abstracción. Así, el código resultante es difícil de leer, escribir, depurar y mantener; y se aleja en forma del formulismo de la teoría estructural.

E. Formulación del Problema

¿Existe un modelo conceptual OO que pueda abstraer y simplificar la complicada codificación del cálculo estructural matricial de pórticos planos mediante MEF capaz de producir resultados completos según la teoría de ejes locales de Kardestuncer?

F. Objetivo General

El objetivo de ésta investigación es proveer y probar un nuevo modelo OO para que el software de análisis estructural de pórticos

planos mediante elementos finitos – elemento viga y sistema de ejes locales Kardestuncer, sea fácil de entender, implementar, extender y modificar.

G. Objetivos Específicos

- Diseñar el modelo OO usando el lenguaje unificado de modelado UML.
- Implementar el modelo OO en el lenguaje de programación Java.
- Resolver un caso de estudio y verificar los resultados

H. Hipótesis

El diseño de un modelo OO para la codificación de un programa para el análisis estructural de pórticos planos mediante elementos finitos – elemento viga y ejes locales Kardestuncer, involucra mayor claridad de código, minimiza la cantidad de ciclos anidados explícitos de cálculo, incrementa la eficiencia de cómputo, mejora los resultados de las fuerzas internas en los elementos y permite escalabilidad.

Independiente	Moderante	Dependiente
El diseño de un modelo OO para la codificación de un programa para el análisis estructural de pórticos planos mediante elementos finitos – elemento viga y ejes locales Kardestuncer.	Mejora	Claridad del código.
	Minimiza	La cantidad de ciclos anidados explícitos de cálculo.
	Incrementa	La eficiencia de cómputo.
	Mejora	Los resultados de las fuerzas internas en los elementos.
	Permite	Escalabilidad.

I. Justificación

La justificación es teórica porque con un modelo OO se pretende desarrollar un aspecto específico de la ingeniería estructural como el análisis de estructuras por elementos finitos – elemento viga y ejes locales Kardestuncer.

La justificación tiene impacto social porque se pretende dejar a disposición de estudiantes avanzados, profesionales e investigadores de ingeniería estructural y mecánica, una técnica (materializada en herramienta) para un análisis eficiente y completo de estructuras planas; las mismas que, tienen un rol de servicio importante en la sociedad.

J. Alcances y Limitaciones

El alcance de ésta investigación es exploratorio porque se pretende indagar el tema desde una nueva perspectiva y porque se desea ampliar una área existente.

El modelo OO que llegue a diseñarse se limitará al análisis estático lineal de estructuras en dos dimensiones, elemento viga

doblemente empotrada, cargas puntuales en los nudos, cargas distribuidas sobre los elementos y apoyos fijos.

La implementación del modelo OO será de carácter inicial y destinado a computadoras personales con el complemento de la máquina virtual de Java instalado. Soportará entrada y salida de datos mediante archivos separados por comas extensión CSV. Ofrecerá una gráfica básica tanto de la geometría de la estructura como de su deformada. La aplicación estará destinada a usuarios avanzados, profesionales o investigadores de ingeniería estructural y mecánica.

II. ESTADO DEL ARTE

[Archer 2000] diseñó e implementó un sistema para el MEF. Su diseño describe la abstracción para cada clase y especifica su interfaz para tal abstracción. Se verifica que la implementación de cada clase y la interacción entre objetos cumplan con la interfaz definida. Además reconoce que el desarrollo de elementos infinitos involucra la adición de nuevos elementos, nuevas estrategias de solución o nuevos esquemas de almacenamiento matriciales; razón por la cual,

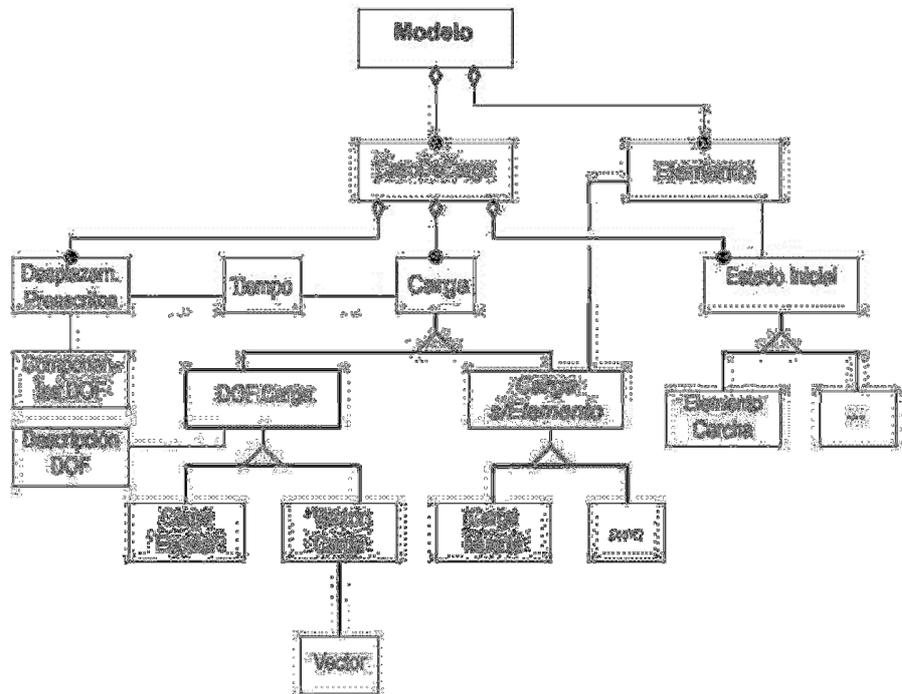
MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

presta especial cuidado a hacer las interfaces lo más flexibles posibles. La transformación entre sistemas de coordenadas es muy

tomada en cuenta. Véase la figura 8. Recuerde que DOF se traduce como grados de libertad (degrees of freedom en inglés).

Archer planteó un modelo OO generalizado para elementos finitos. En ésta investigación se propone un modelo OO específico - elemento viga con un sistema de ejes locales Kardestuncer.

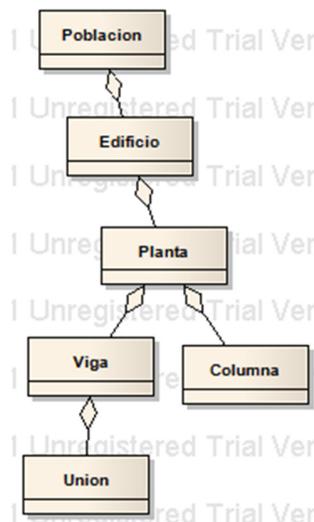
FIGURA 8. MODELO ORIENTADO A OBJETOS PROPUESTO POR ARCHER.



Dicha arquitectura OO está dirigida a aplicaciones de análisis lineal elástico o análisis no lineal.

Jun Lu desarrolló una plataforma OO universal para el análisis estructural por elementos finitos. En ésta investigación se desarrollará una plataforma OO específica para el análisis estático de pórticos 2D por elementos finitos – elemento viga y ejes locales Kardestuncer.

FIGURA 9. MODELO OO DE FOLEY PAR ESTRUCTURAS DE ACERO



[Al Vermeulen 2000] describió las ventajas de los lenguajes OO como el C++ con relación a los lenguajes de programación basados en procedimientos como el FORTRAN. Expuso el código desarrollado en su investigación basado en un elemento finito de curva cúbica de deformación. Ejemplificó el método de los elementos finitos explicando los esfuerzos y deformaciones que sufre la rueda de una bicicleta. A raíz de que el elemento finito de curva cúbica de deformación conduce a ecuaciones complejas, concluyó que la programación OO es la más recomendable para lidiar mejor con ésta complejidad. Al Vermeulen implementó en C++ un modelo OO para elementos finitos de curva cúbica de deformación. En ésta investigación se implementará en Java un modelo OO para elementos finitos de deformación lineal - elemento viga y ejes locales Kardestuncer.

MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

[Foley 2003] propuso un algoritmo de computación evolutiva para el diseño de pórticos de acero, parcial o totalmente restringidos. Aplicó un análisis avanzado de diseño estructural usando un modelo inelástico basado en el comportamiento plástico distribuido. La automatización del proceso de diseño estructural es implementada usando un algoritmo evolutivo OO. Empleó la orientación a objetos OO para modelar los componentes del pórtico y las operaciones de reproducción de dicho algoritmo evolutivo. Demostró su propuesta diseñando pórticos de ejemplo de diversa topología y discutió el rendimiento resultante de su algoritmo evolutivo OO.

Foley propuso un algoritmo evolutivo OO para el diseño de pórticos de acero. En ésta investigación se propone un modelo OO para el análisis de pórticos por elementos finitos.

[Bittencourt et al 2001] desarrolló en C++ un software OO interactivo para el análisis y optimización estructural en dos dimensiones. La principal característica de dicho software es la integración de varias herramientas de cálculo en una interfaz gráfica de usuario. Ésta interfaz simplifica la introducción de datos para la simulación y optimización de problemas elásticos lineales de dos dimensiones. La implementación fue realizada para el sistema operativo Windows-NT. [Bittencourt desarrolló en C++ una aplicación para el análisis y optimización estructural 2D. En ésta investigación se desarrollará en Java una aplicación solo para el análisis estructural 2D – elemento viga y ejes locales Kardestuncer.

[McKenna 2000] diseñó un modelo OO para una variedad métodos de análisis por elementos finitos usando computación secuencial y paralela. En un diseño tradicional OO se crea un objeto único para realizar el análisis. Pero el diseño de McKenna propone una variedad de subclases para una clase “Análisis” que modela distintos análisis como ser: Integrator, ConstraintHandler, DOF_Numberer, AnalysisAlgorithm y otros. Este enfoque ofrece una gran flexibilidad al permitir elegir el análisis mediante el tipo de objeto pasado al constructor de clase. El modelo fue implementado

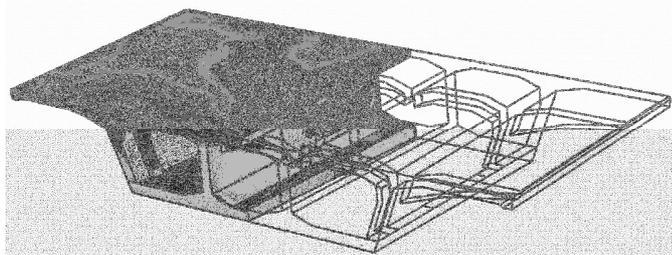
en C++ y comparado con un código en un lenguaje basado en procedimientos. McKenna diseñó un modelo OO con una clase “Análisis” segmentada en subclases para varios tipos de análisis por elementos finitos. En ésta investigación se diseñará un modelo OO que involucra una clase “Análisis” no-segmentada y específica para el análisis estructural estático de pórticos 2D.

[Patzak 2001] creó un código gratuito de arquitectura OO para la solución de problemas de mecánica, transporte y fluidos; ejecutable en varias plataformas. Su objetivo fue desarrollar una herramienta eficiente y robusta para el MEF, y proveer de un ambiente modular y extensible para desarrollos posteriores. El código incluye un módulo para el balance dinámico de carga de trabajo en ambientes de computación paralela. Véase figura 11. Patzak desarrolló un código OO para resolver problemas de mecánica, transporte y fluidos mediante elementos finitos. En ésta investigación se desarrollará un código OO para resolver problemas de mecánica; específicamente, para el análisis de pórticos 2D por elementos finitos – elemento viga y ejes locales Kardestuncer.

[Biniaris 2004] desarrolló una aplicación OO para resolver problemas 3D por el MEF aplicando computación distribuida. La implementó en una red de estaciones de trabajo. La solución del sistema de ecuaciones resultantes se la lleva a cabo de forma distribuida por el método de la gradiente conjugada. La implementación fue en Java y la computación distribuida se logró mediante el uso de la tecnología agente móvil. El punto principal de dicha investigación está dedicado a una descripción de la implementación del modelo OO distribuido. Presentó resultados que prueban la ventaja de la técnica. Véase figura 10.

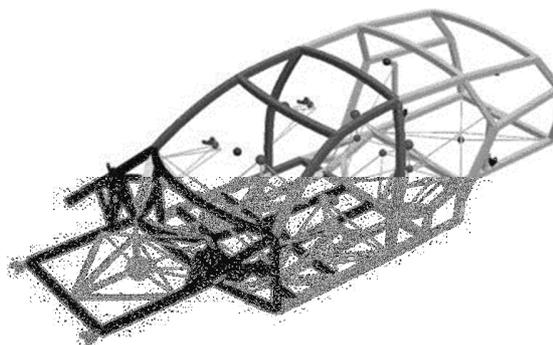
Biniaris desarrolló en Java un modelo OO para la solución de problemas 3D por elementos finitos aplicando computación distribuida. En ésta investigación se desarrollará un modelo OO para la solución de problemas estructurales 2D por elementos finitos aplicando computación secuencial.

FIGURA 10. SECCIÓN DE PUENTE. GRÁFICA DE LOS ESFUERZOS LUEGO DEL ANÁLISIS MEDIANTE EL MEF.



[VectorSpaceProgramming 2008] desarrolló en C++ una librería matemática con un conjunto de objetos matemáticamente integrables

FIGURA 11. MODELO DE ELEMENTOS FINITOS DE UN VEHÍCULO.



y diferenciables. La librería es una herramienta para desarrollar rápidamente aplicaciones de cálculo.

MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

La librería está dirigida a aplicaciones OO para el FEM que involucren análisis numérico avanzado, programación lineal, optimización, métodos de las diferencias finitas y métodos variacionales. Se trata de una librería en C++ para programas OO sobre el FEM que además dispone de métodos numéricos avanzados. En ésta investigación se implementará en Java una librería OO específica para problemas estructurales mediante elementos finitos. No se programarán métodos numéricos avanzados.

III. DESARROLLO TEÓRICO PRÁCTICO - DECLARACIÓN DE TRABAJO

Un elemento puede tener varias condiciones de carga puntual y/o distribuida en común con otros elementos; sin embargo, una condición de carga debe pertenecer solo a un elemento. Un nudo puede tener solo un conjunto de coordenadas planas (X, Y). Y un conjunto de coordenadas planas puede pertenecer únicamente a un nudo. Un nudo puede tener varias solicitaciones de carga puntual en común con otros nudos pero una condición de carga puntual debe pertenecer solo a un nudo.

Cada nudo puede tener solo un conjunto de condiciones de borde o de frontera. Un conjunto de condiciones de borde pueden tener en común varios nudos. Un nudo puede tener solo un conjunto de desplazamientos. Un conjunto de desplazamientos pueden tener en común más de un nudo. Un nudo restringido (apoyo) puede tener solo un conjunto de reacciones. Y un conjunto de reacciones pueden tener en común varios nudos.

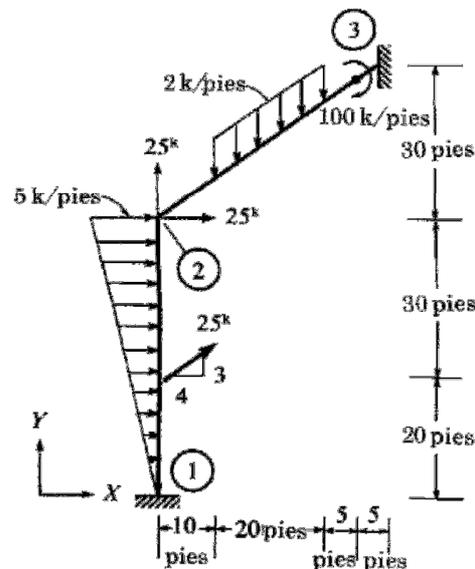
Una estructura (figura 12) se compone de nudos y elementos (elemento viga). Cada elemento parte de un solo nudo y termina en un solo nudo. Puede haber elementos que parten de un mismo nudo o que terminan en un mismo nudo. Un elemento puede tener solo un conjunto de propiedades de la sección (módulo de elasticidad, área y momento de inercia). Puede haber varios elementos que tengan las mismas propiedades de la sección. Un elemento puede tener solo un conjunto de fuerzas internas (fuerza axial, fuerza cortante y momento flector). Un conjunto de fuerzas internas puede pertenecer a más de un elemento.

La información de entrada de una estructura consiste en: una descripción de la estructura, las coordenadas globales (X, Y) de sus nudos, las incidencias de sus elementos, un listado de las cargas puntuales sobre sus nudos, un listado de las cargas puntuales y/o distribuidas sobre sus elementos, y un listado de las condiciones de contorno (restricciones o apoyos).

La información que se requiere luego del cálculo de una estructura consiste en: la matriz global de rigidez, el vector generalizado de fuerza, el vector generalizado de desplazamiento, las fuerzas internas en los extremos de los elementos según el sistema de ejes locales Kardestuncer, las reacciones en los apoyos y una gráfica de la deformada de la estructura.

El cálculo de una estructura deberá llevarse a cabo por medio del método de los elementos finitos empleado en ingeniería estructural conforme el extenso desarrollo de la sección "marco teórico" de éste informe.

FIGURA 12. UNA ESTRUCTURA 2D DE 3 NUDOS, 2 ELEMENTOS, 3 CARGAS PUNTUALES, 2 CARGAS DISTRIBUIDAS Y 2 APOYOS FIJOS.



A. *Requerimientos*

- 1 Al usuario se le deberá proveer de un formato de archivo estándar que permita la introducción de manera tabulada de la información de entrada de la estructura.
- 2 Al usuario se le deberá proveer un campo que permita la introducción de la descripción literal de la estructura.
- 3 Al usuario se le deberá proveer de un campo que permita la introducción de las coordenadas (X, Y) de los nudos de la estructura.
- 4 Al usuario se le deberá proveer de un campo que permita la introducción de las incidencias, de las propiedades geométricas y de las propiedades inerciales de los elementos de la estructura.
- 5 Al usuario se le deberá proveer de campo que permita la introducción de las cargas puntuales aplicadas en los nudos de la estructura.
- 6 Al usuario se le deberá proveer de un campo que permita la introducción de las cargas puntuales y/o distribuidas que actúan sobre los elementos de la estructura.
- 7 Al usuario se le deberá proveer de un campo que permita la introducción de las condiciones de borde (apoyos) de la estructura.
- 8 Al usuario se le deberá proveer una interfaz para la verificación gráfica de la geometría de la estructura.
- 9 Al usuario se le deberá proveer una interfaz para solicitar el análisis de la estructura.
- 10 Al usuario se le deberá proveer archivos de un formato estándar que permita acceder de manera tabulada a los resultados numéricos del análisis de la estructura.
- 11 Al usuario se le deberá proveer de un archivo para acceder a los resultados numéricos de la matriz global de rigidez de la estructura.
- 12 Al usuario se le deberá proveer de un archivo para acceder a los resultados numéricos del vector generalizado de fuerza de la estructura.
- 13 Al usuario se le deberá proveer de un archivo para acceder a los resultados numéricos del vector generalizado de desplazamiento de la estructura.
- 14 Al usuario se le deberá proveer de un archivo para acceder a los resultados numéricos de las fuerzas internas en los extremos de los elementos de la estructura.
- 15 Al usuario se le deberá proveer de un archivo para acceder a los resultados numéricos de las reacciones en los apoyos de la estructura.
- 16 Al usuario se le deberá proveer de una interfaz gráfica para verificar la deformada resultante del análisis de la estructura.
- 17 Al sistema se le deberá proveer la implementación de un modelo que permita el análisis de la estructura mediante el método de los elementos finitos - elemento viga doblemente empotrada sistema de ejes locales Kardestuncer según el extenso desarrollo de la sección “marco teórico” de éste documento.
- 18 Al sistema se le deberá proveer una rutina para abrir la información de entrada de la estructura.
- 19 Al sistema se le deberá proveer una rutina para graficar la geometría de la estructura.
- 20 Al sistema se le deberá proveer una rutina para ensamblar el vector generalizado de fuerza de la estructura.
- 21 Al sistema se le deberá proveer una rutina para ensamblar la matriz global de rigidez la estructura.
- 22 Al sistema se le deberá proveer una rutina para aplicar las condiciones de contorno de la estructura.
- 23 Al sistema se le deberá proveer una rutina para resolver el sistema de ecuaciones de la estructura.
- 24 Al sistema se le deberá proveer una rutina para calcular las fuerzas internas en los extremos de los elementos de la estructura.
- 25 Al sistema se le deberá proveer una rutina para calcular las reacciones de los apoyos de la estructura.
- 26 Al sistema se le deberá proveer una rutina para guardar los resultados numéricos del análisis de la estructura.
- 27 Al sistema se le deberá proveer una rutina para graficar la deformada de la estructura.

B. *Modelo de Casos de Uso*

Los actores que se distinguen del análisis de requerimientos son “el usuario” y “el sistema” según se muestran en los casos de uso de la figura 13. En general, es posible detallar con mayor profundidad la mayoría de los casos

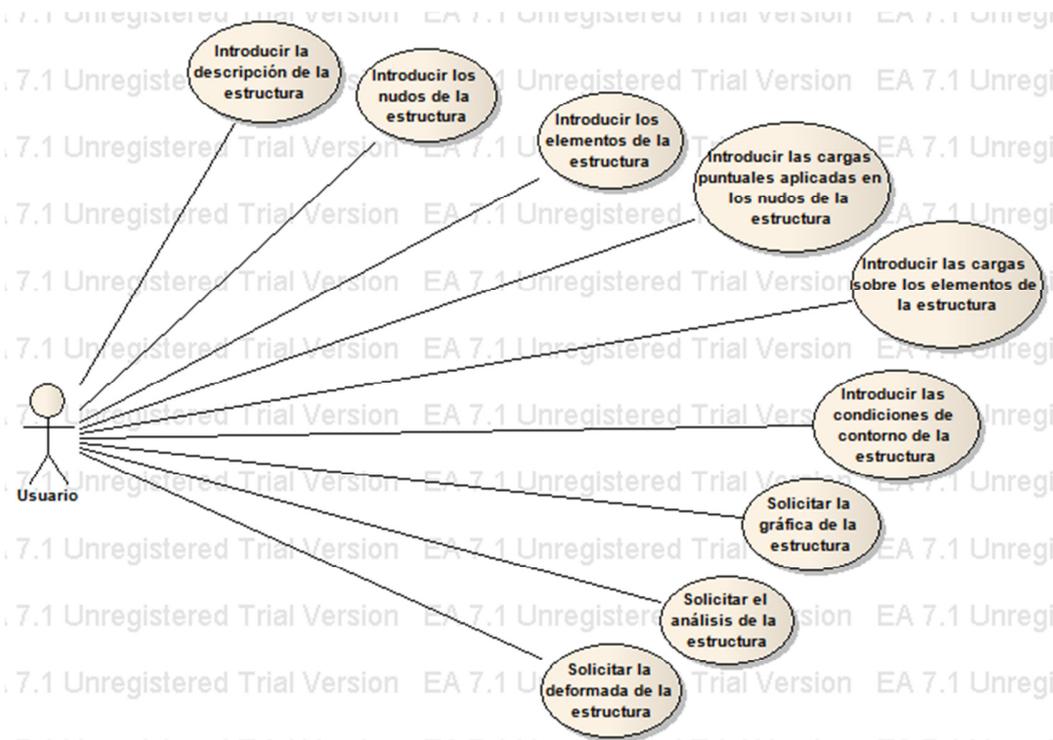
uso; a este concepto se conoce como “inclusión de caso de uso”.

C. Modelo de Clases

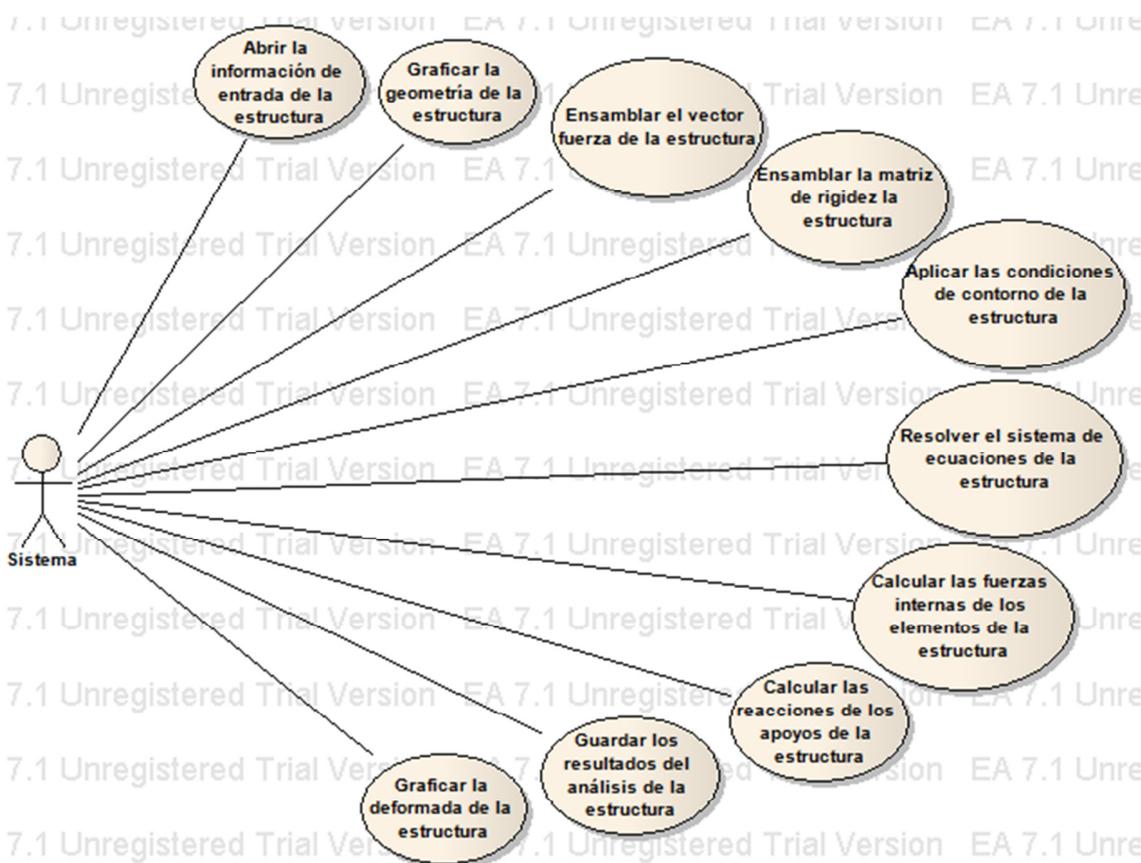
Del análisis de requerimientos, en general, las clases se corresponden con los sustantivos que a su vez son objetos: estructura, elementos, nudos, cargas, y otros. Por otra parte, los verbos casi siempre se correlacionan con las

responsabilidades (operaciones) de los objetos como ser: ensamblar sistema de ecuaciones, aplicar condiciones de contorno, resolver sistema de ecuaciones, calcular fuerzas internas, calcular reacciones y otros.

FIGURA 13. CASOS DE USO – ACTOR “EL USUARIO” Y ACTOR “EL SISTEMA”.



MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS



Luego de una serie de revisiones, se ha llegado a un diseño de clases robusto (figura 14) basado en la encapsulación, herencia, abstracción, polimorfismo, composición, clases abstractas, interface java, y otros. En cada clase se aplica la encapsulación ocultando las propiedades y permitiendo el acceso a las mismas mediante métodos y su interfaz. Se emplea la herencia para especializar las clases abstractas y crear clases que se puedan instanciar para resolver casos específicos. Gracias a un adecuado conocimiento del dominio se ha llegado a un alto grado de abstracción del modelo dando lugar a futura escalabilidad (e. g. estructuras 3D). El polimorfismo es un concepto empleado sobre todo en las subclases del modelo. Se emplea la composición por agregación pero ésta puede distinguirse en la zona “propiedades” de cada clase. Se emplea el concepto de interface java como artificio para proveer de herencia múltiple al modelo ya que en Java no se tiene herencia múltiple.

La interface java interface01 se emplea para que cada clase del modelo esté obligada a implementar un método de identificación para cualquier objeto. Nótese que las clases que “realizan” a la interface java “interface01” deben ser abstractas.

La clase abstracta estructura00 contiene todas las propiedades y métodos fundamentales de una estructura en general o sea ésta superclase podría aplicarse sin ningún inconveniente inclusive a

estructuras 3D. Nótese que muchas de las operaciones de ésta superclase tienen que ver directamente con el análisis estructural mediante elementos finitos.

La clase abstracta nudo00 contiene todas las propiedades y métodos fundamentales de un nudo en general; sin embargo, de ésta superclase podría heredarse otros tipos de nudos como nudos con apoyos elásticos u otros.

La clase abstracta elemento00 abstrae las propiedades y métodos para un elemento estructural en general; ésta superclase podría aplicarse inclusive como base de elementos 3D. Para las estructuras planas como los marcos planos, es necesario heredar una subclase como elemento01 que permita la especialización en cuanto a la matriz de rigidez, matriz de rotación, matriz de coeficientes de fuerza y otros para los elementos de dichas estructuras. La subclase elemento02 es un adelanto de prueba del modelo; se trata de la especialización para elementos de otras estructuras planas como grillas utilizadas en losas, puentes, y otros.

La clase abstracta cargae00 abstrae las propiedades y métodos para las cargas puntuales o distribuidas aplicadas sobre los elementos de una estructura. Se ha heredado, especializado y llegado a una subclase cargae01 para las cargas puntuales aplicadas sobre un elemento. La subclase cargae02 es una herencia y especialización

de `cargaele00` para el caso de cargas distribuidas rectangulares aplicadas sobre el elemento. Para la siguiente revisión se va a incluir una subclase especializada `cargaele03` para cargas distribuidas triangulares aplicadas sobre un elemento (véase figura 12).

Finalmente, las subclases `estructura01` y `nudo01` son clases heredadas que permiten instanciarse para la aplicación práctica de sus correspondientes clases abstractas. Las subclases `estructura01` y `nudo01` también tienen la función de la inicialización de las propiedades heredadas y de la identificación que exige la realización de la interface java `interface01`.

D. Vista Dinámica – Diagrama de Secuencia

Basado en el algoritmo del método de los elementos finitos para el análisis de estructuras, se ha diseñado la comunicación entre los objetos del modelo en función del tiempo según se muestra en el diagrama de secuencia para el caso de uso “ensamblar sistema de ecuaciones de la estructura” cuyo actor es “el sistema” (véase figura

15). El algoritmo exige el ensamblado de la matriz global de rigidez de la estructura para cuyo efecto se calculan primero las rigideces locales y globales de los elementos. El MEF también exige el ensamblado del vector generalizado de fuerza de la estructura para lo cual se consideran las cargas sobre los elementos y las cargas sobre los nudos que pueda haber. En el diagrama se pueden verificar que todos los mensajes entre objetos son sincrónicos.

E. Diagrama de Componentes

Además del paquete de clases para el análisis de estructuras mediante el MEF que se propone en ésta investigación y que se denomina “Estructural”, en el diagrama de componentes de la figura 16, intervienen otras clases como herramientas para que la aplicación resultante pueda cumplir con la lista de requerimientos.; es el caso del paquete JAMA de distribución gratuita que viene con un modelo OO para las operaciones del álgebra matricial incluyendo métodos de factorización para la resolución de ecuaciones matriciales como las que forman las estructuras.

FIGURA 14. MODELO DE CLASES.

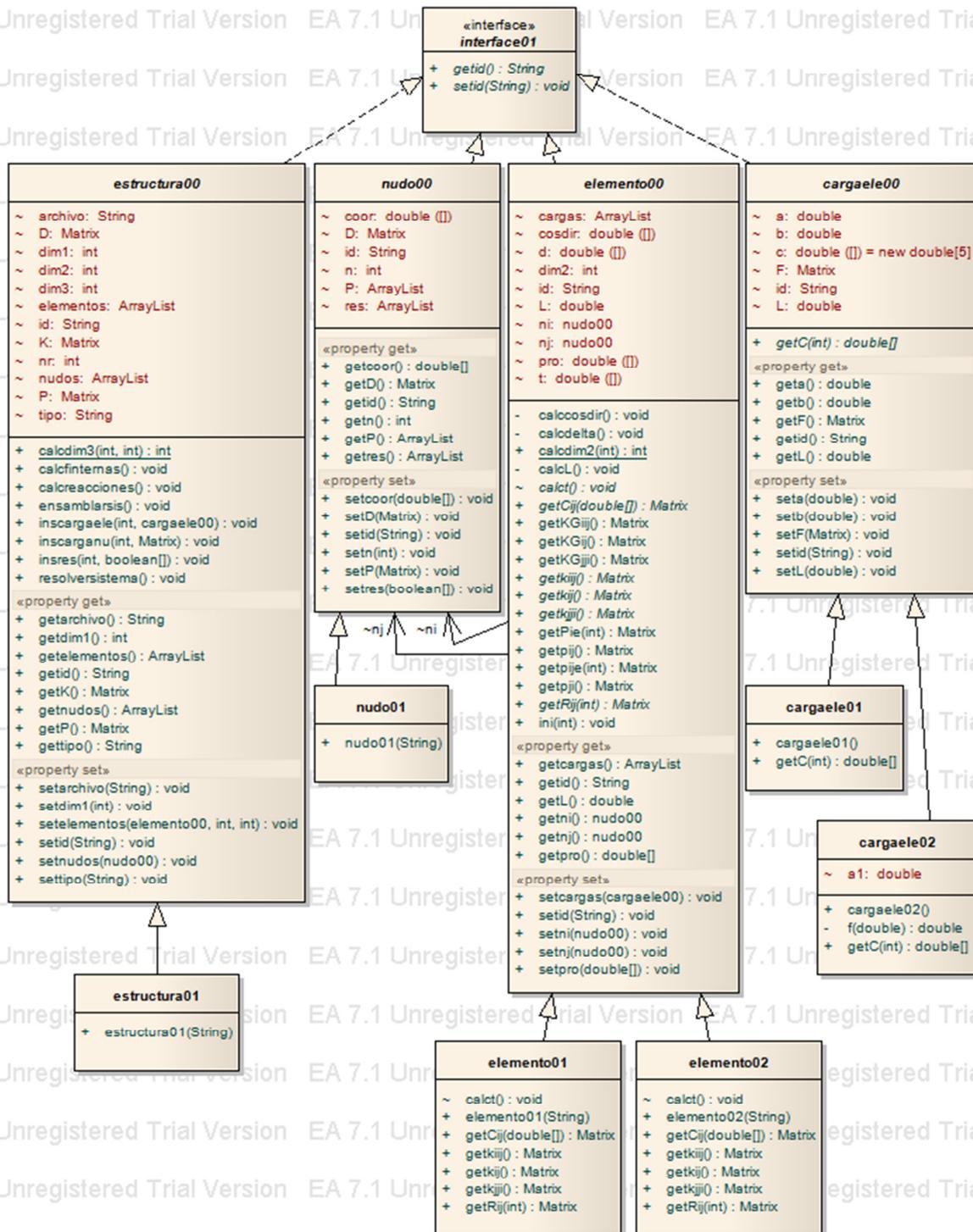


FIGURA 15. DIAGRAMA DE SECUENCIA PARA EL CASO DE USO "ENSAMBLAR EL SISTEMA DE ECUACIONES".

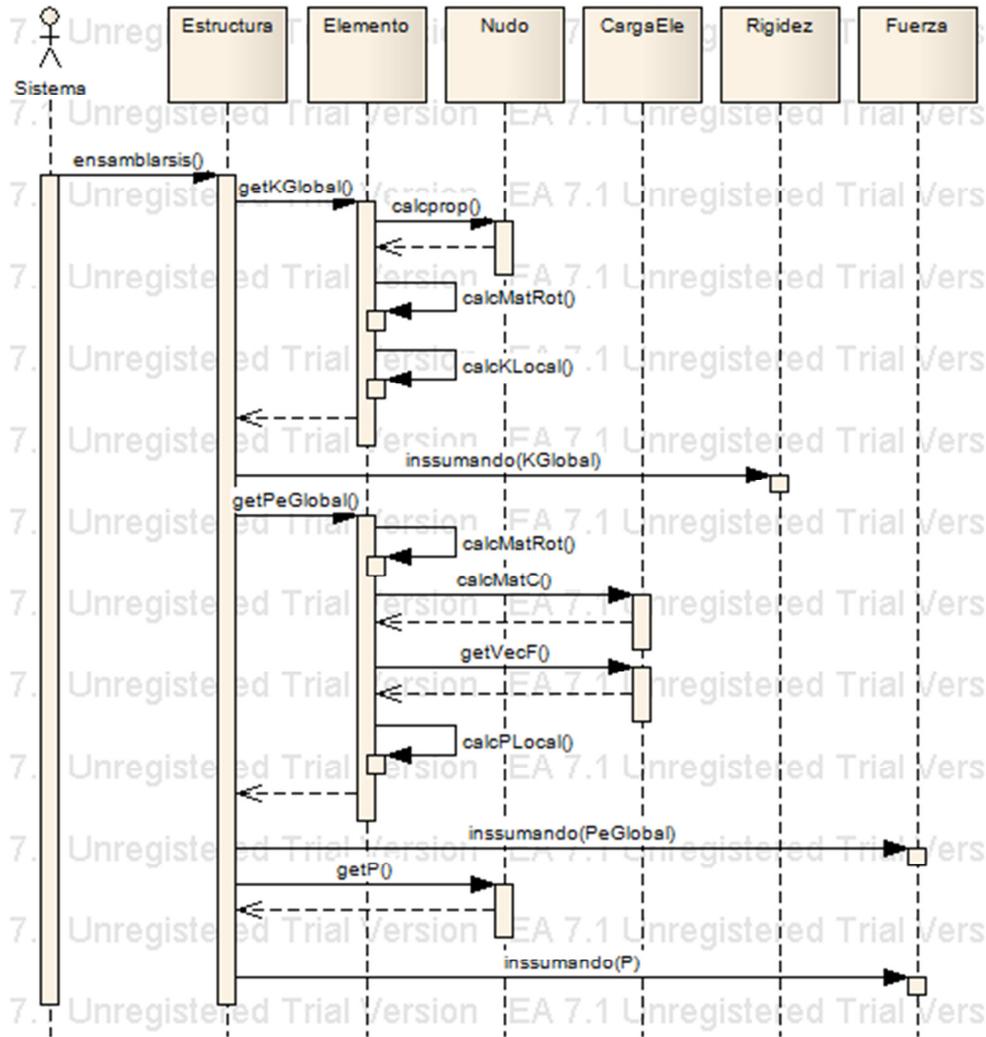
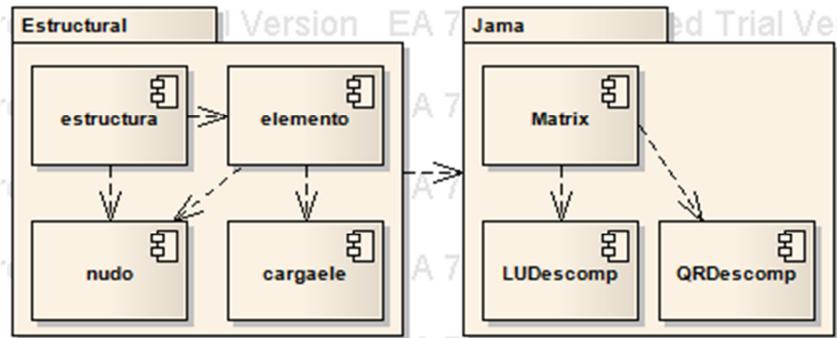


FIGURA 16. DIAGRAMA DE COMPONENTES



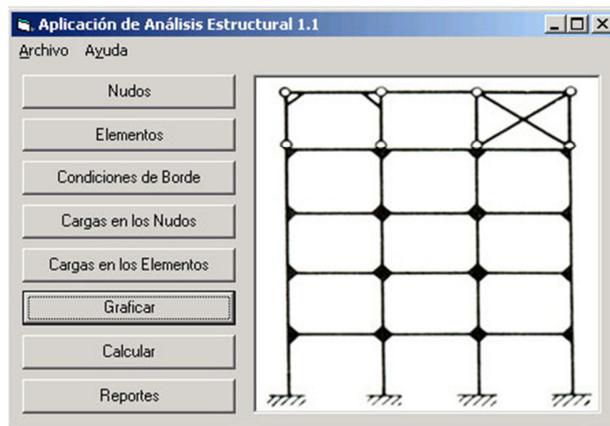
F. Implementación – Generalidades

La implementación del modelo de clases de la figura 17 se la ha llevado a cabo en Java. Todas las clases han sido codificadas con base a la ventaja fundamental de la programación OO: la encapsulación. El ocultamiento de las propiedades y métodos ha sido aplicado a todas las clases. El concepto de “getters” y “setters” para proveer acceso

controlado a los datos de un objeto ha sido aplicado en todos los casos para evitar que haya manipulación directa de datos (atributos) entre objetos.

El prototipo de la interfaz de usuario de la aplicación se muestra en la figura 17. El prototipo proporcionará información que ayudará a mejorar la aplicación durante las iteraciones del proceso de diseño.

FIGURA 18. CASO DE ESTUDIO



G. Caso de Estudio

Se va a emplear el programa para analizar una estructura tipo cercha. Y para probar el modelo, se va a verificar manualmente el equilibrio primero entre las fuerzas internas que concurren a un nudo elegido al azar; luego, entre fuerzas externas versus reacciones en los apoyos.

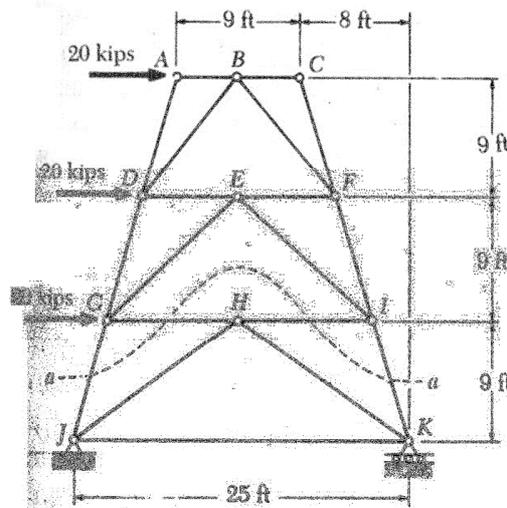
Fuente: BEER Y JOHNSTON (1993). “Mecánica Vectorial para Ingenieros”. Editorial McGraw-Hill. México. Páginas 243 y 469.

Problema 6.40: Calcular la fuerza en el miembro I-K de la cercha plana que se muestra en la figura 18.

Solución: Se crea el archivo de entrada con un software para planillas electrónicas como Microsoft Excel.

Primero, se introduce un metadato DESCRIPCION DE LA ESTRUCTURA y luego un título para la estructura como “BEER Y JOHNSTON Páginas 243 y 469 Problema 6.40”.

FIGURA 18. CASO DE ESTUDIO



Segundo, se introduce un metadato COORDENADAS DE LOS NUDOS; luego, se elige un origen para el sistema global de coordenadas XY de la estructura como ser en el nudo J. Y se introducen las coordenadas X e Y de los nudos A (nudo 1) hasta la J (11). Tercero, se introduce un metadato ELEMENTOS TIPO y abajo se coloca "portico" para indicar que los elementos de la estructura se corresponden con un marco plano. La estructura de la figura 18 es una cercha que puede tratarse como pórtico de elementos de inercia cero. Cuarto, se ingresa un metadato PROPIEDADES DE LOS ELEMENTOS; seguidamente, se introducen las incidencias de los elementos, el módulo de elasticidad, el momento de inercia y el área de la sección. Quinto, se entra un metadato CARGAS EN LOS NUDOS y luego las cargas puntuales aplicadas en los nudos A (nudo 1), D (4) y G (7) de la figura. Siempre se introduce primero la componente horizontal de la carga, luego la vertical y finalmente el momento flector. Sexto, entrar un metadato CARGAS EN LOS ELEMENTOS para cumplir 1 formato de archivo exigido por el programa ya que ésta estructura no tiene cargas sobre elementos. Séptimo, ingrese un metadato APOYOS para describir las condiciones de contorno de la estructura. Abajo colóquese primero el número de nudo que corresponde al apoyo, luego la restricción al desplazamiento horizontal del nudo, al desplazamiento vertical y finalmente la restricción al giro. Cuando exista restricción en alguna dirección solo se coloca 1 (uno) y 0 (cero) cuando no. Nótese que para el problema en cuestión, los nudos J (nudo 10) y K (nudo 11) están parcialmente restringidos.

La figura 19 muestra cómo debiera quedar la introducción de la estructura. Finalmente, guárdese la hoja electrónica en formato de archivo de valores separados por comas o CSV (commaseparatedvalues). Elíjase un nombre que represente a la estructura porque luego los nombres de los archivos CSV resultantes del análisis estructural se basarán en el nombre del archivo de entrada.

Al ejecutar la aplicación, ésta recupera el archivo CSV de entrada de la estructura y en caso de que la información esté correcta, muestra una gráfica para la verificación de la geometría de la estructura. Una captura de pantalla se muestra en la figura 20.

A continuación, la aplicación realiza el análisis de la estructura y se van guardando archivos CSV de resultados numéricos importantes a medida del progreso del proceso.

Cuando concluye el análisis aparece una ventana con una gráfica exagerada de la deformada de la estructura (figura 21). Nótese que la deformada coincide con la tendencia de desplazamiento de la estructura bajo las condiciones de carga horizontal aplicada.

Cada ecuación del sistema de ecuaciones simultáneas de la estructura, corresponde a cada grado de libertad de la misma. Esta estructura de 11 nudos donde cada nudo tiene 3 grados de libertad (i. e. desplazamiento horizontal, vertical y rotacional) forma un sistema de 33 ecuaciones (33 incógnitas).

La matriz de rigidez de la estructura se muestra en la página siguiente. La matriz de rigidez K es cuadrada, tiene 33 filas, 33 columnas y es simétrica conforme una de sus propiedades. Todo indica que ha sido ensamblada correctamente por el sistema.

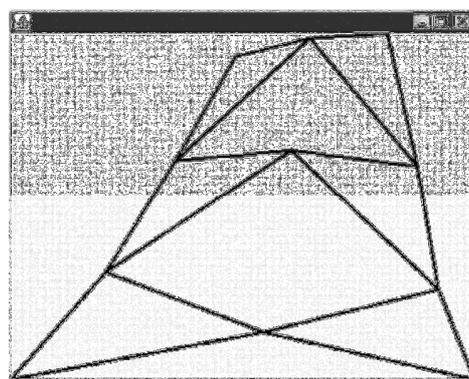
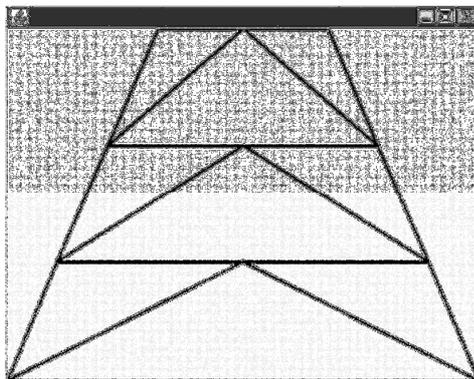
El fin es diseñar una estructura y el análisis estructural es una herramienta para alcanzar tal fin. Los resultados más importantes del análisis son: las fuerzas internas en los extremos de los elementos y las reacciones en los apoyos. El modelo OO implementado en éste programa mejora los resultados del análisis estructural por medio de un elemento finito viga de ejes locales Kardestuncer; razón por la cual, se tienen valores de las fuerzas internas para ambos extremos de los elementos según se ve en la figura 22.

FIGURA 19. ENTRADA DE LA ESTRUCTURA.

	A	B	C	D	E
1	DESCRIPCION DE LA ESTRUCTURA				
2	BEER Y JOHNSTON Páginas 243 y 468 Problema 6.40				
3					
4	COORDENADAS DE LOS NUDOS				
5	8	27			
6	12.5	27			
7	17	27			
8	5.0000000	19			
9	12.5	19			
10	19.0000007	19			
11	2.0000007	3			
12	12.5	3			
13	22.0000009	3			
14	0	9			
15	26	9			
17	ELEMENTOS TIPO				
18	portico				
19					
20	PROPIEDADES DE LOS ELEMENTOS				
21	1	2	1	1.00E-09	1
22	1	4	1	1.00E-09	1
23	2	3	1	1.00E-09	1
24	2	4	1	1.00E-09	1
25	2	6	1	1.00E-09	1
26	3	8	1	1.00E-09	1
27	4	5	1	1.00E-09	1
28	4	7	1	1.00E-09	1
29	5	6	1	1.00E-09	1
30	5	7	1	1.00E-09	1
31	5	9	1	1.00E-09	1
32	6	9	1	1.00E-09	1
33	7	9	1	1.00E-09	1
34	7	10	1	1.00E-09	1
35	8	9	1	1.00E-09	1
36	8	10	1	1.00E-09	1
37	8	11	1	1.00E-09	1
38	9	11	1	1.00E-09	1
39	10	11	1	1.00E-09	1
40					
41	CARGAS EN LOS NUDOS				
42	1	20	0	0	
43	4	20	0	0	
44	7	20	0	0	
45					
46	CARGAS EN LOS ELEMENTOS				
47					
48	APOYOS				
49	10	1	1	0	
50	11	0	1	0	

Figura 20. Geometría de la estructura antes del análisis.

Figura 21. Deformada de la estructura luego del análisis.



MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

Se tienen fuerzas internas solo en el eje local axial x (Px); esto coincide con el comportamiento de una cercha (figura 22).

Las fuerzas internas del elemento I-K (nudos 9-11) elemento 18:

$$\begin{matrix}
 \text{Extremo i} & & \text{Extremo j} \\
 F_{9-11} = \begin{bmatrix} -28.64 \\ 0.00 \\ 0.00 \end{bmatrix} & & F_{11-9} = \begin{bmatrix} -28.64 \\ 0.00 \\ 0.00 \end{bmatrix}
 \end{matrix}$$

Un signo negativo indica compresión en el extremo del elemento según los ejes locales de Kardestuncer (figura 4b).

La respuesta textual de la fuente bibliográfica es:

6.40 28.6 kips C.

La C indica compresión. Entonces, la respuesta del programa es correcta.

Los resultados para las reacciones en los apoyos son:

Apoyo	Rx	Ry	Mz
10.00	-60.00	-43.20	0.00
11.00	0.00	43.20	0.00

La suma de componentes verticales Ry da 0 (cero) lo cual indica que no existe componente vertical de carga externa o sea es correcto.

Verificación Manual de Resultados

La verificación manual del análisis estructural consiste en:
 1) Elegir un nudo al azar y realizar la comprobación de equilibrio de las fuerzas internas/externas que concurren al nudo.
 2) Realizar la sumatoria de las fuerzas externas aplicadas incluyendo las reacciones. Para verificar el equilibrio la suma debe dar 0 (cero).

Concurren al nudo D	Valor en el nudo	Horizontal	Vertical
Elemento AD (2)	0.00	0.00	0.00
Elemento BD (4)	+16.05	+10.00	+12.56
Elemento ED (7)	-26.28	-26.28	0.00
Elemento DG (8)	-13.10	-3.72	-12.56
Carga externa	+ 20.00	+20.00	0.00
Sumatoria		0.00	0.00

Se verifica exitosamente el equilibrio en el nudo D de la estructura lo cual indica que el modelo y su implementación (i. e. el programa) están produciendo resultados correctos para las fuerzas internas en los extremos de los elementos.

FIGURA 22. FUERZAS INTERNAS EN LOS EXTREMOS DE LOS ELEMENTOS.

Elemento	Pxi	Pyi	Mzi	Pxj	Pyj	Mzj
1.00	-20.00	0.00	0.00	-20.00	0.00	0.00
2.00	0.00	0.00	0.00	0.00	0.00	0.00
3.00	0.00	0.00	0.00	0.00	0.00	0.00
4.00	16.05	0.00	0.00	16.05	0.00	0.00
5.00	-16.05	0.00	0.00	-16.05	0.00	0.00
6.00	0.00	0.00	0.00	0.00	0.00	0.00
7.00	-26.28	0.00	0.00	-26.28	0.00	0.00
8.00	13.10	0.00	0.00	13.10	0.00	0.00
9.00	6.28	0.00	0.00	6.28	0.00	0.00
10.00	22.07	0.00	0.00	22.07	0.00	0.00
11.00	-22.07	0.00	0.00	-22.07	0.00	0.00
12.00	-13.10	0.00	0.00	-13.10	0.00	0.00
13.00	-31.86	0.00	0.00	-31.86	0.00	0.00
14.00	28.64	0.00	0.00	28.64	0.00	0.00
15.00	11.86	0.00	0.00	11.86	0.00	0.00
16.00	26.94	0.00	0.00	26.94	0.00	0.00
17.00	-26.94	0.00	0.00	-26.94	0.00	0.00
18.00	-28.64	0.00	0.00	-28.64	0.00	0.00
19.00	30.00	0.00	0.00	30.00	0.00	0.00

Por motivos de ilustración procedimental, el nudo elegido para comprobar el equilibrio puede ser el nudo D (nudo 4). Se puede usar una tabla para recopilar, ordenar, descomponer y sumar las fuerzas que concurren al nudo D:

Ahora bien, la tabla para la verificación del equilibrio de las fuerzas externas y reacciones que actúan en la estructura es la siguiente:

Descripción	Horizontal	Vertical
Carga externa nudo A	+20.00	0.00
Carga externa nudo D	+20.00	0.00
Carga externa nudo G	+20.00	0.00
Reacción apoyo J	-60.00	-43.20
Reacción apoyo K	0.00	+43.20
Sumatoria	0.00	0.00

Se verifica exitosamente el equilibrio de las cargas externas y reacciones aplicadas en los nudos de la estructura. Se ratifica que la implementación del modelo OO para el análisis estructural mediante elementos finitos – elemento viga sistema de ejes locales Kardestuncer, está produciendo resultados correctos.

En las siguientes revisiones de ésta investigación se compararán los resultados con aquellos de aplicaciones tradicionales comerciales como el SAP2000, y otros.

IV. CONCLUSIONES Y RECOMENDACIONES

Las estructuras de ingeniería tienen un rol de servicio importante en el Mundo. Una estructura debe cumplir su misión con resistencia y rigidez.

El análisis estructural se usa para diseñar estructuras. El análisis determina la deformada de la estructura para verificar si es adecuada para sus cargas de diseño.

El método de los elementos finitos MEF es para simular sistemas de ingeniería en computadora. El MEF consiste en: elegir un elemento para el modelado, aplicar un proceso para lograr un modelo global y resolver el sistema de ecuaciones de la estructura.

El método de la rigidez es el proceso que requiere el MEF al tratar con estructuras. El método de la rigidez establece la relación entre las cargas externas y los desplazamientos de una estructura.

Los marcos planos o pórticos son estructuras 2D cuyas cargas externas están en el mismo plano. Las cerchas pueden modelarse como marcos planos manteniendo la inercia de sus elementos casi cero.

Un solo sistema de ejes locales para ambos extremos de un elemento estructural es una simplificación que se ha usado siempre pero induce a ambigüedades en la interpretación de los resultados del análisis. El sistema de ejes locales Kardestuncer es mejor porque provee un sistema de ejes locales para cada extremo de los elementos de una estructura.

Los objetos resuelven los problemas de la programación procedimental al combinar datos y comportamiento en un solo paquete encapsulado.

El objetivo del análisis OO es producir las clases relevantes al problema que se va a resolver. El diseño OO transforma el modelo del análisis en un modelo que sirve para el desarrollo de software.

Debido a las computadoras el MEF ha sido ampliamente como herramienta de modelado y simulación de sistemas en ingeniería. Los programas tradicionales siempre han consistido en miles de líneas de código dificultoso de leer, escribir y mantener en lenguajes procedimentales...

El estado del arte revela nuevos tipos de elementos finitos, esquemas matriciales, modelos generales OO, aplicaciones de análisis no-lineal, algoritmos de computación evolutiva de diseño estructural con modelos OO para análisis plástico no-lineal, optimización estructural, plataformas OO para MEF multipropósito, modelos OO para problemas MEF 3D mediante computación distribuida, librerías matemáticas para aplicaciones OO de MEF, y otros.

El diseño del modelo OO propuesto en ésta investigación emerge de una declaración de trabajo a partir de la cual se realiza un análisis de requerimientos para poder diseñar las distintas vistas del modelo OO usando UML.

En los casos de uso se distinguen dos actores: el sistema y el usuario. Del análisis de requerimientos, los sustantivos se convierten en clases y los verbos en operaciones de los objetos. El modelo OO resultante está basado en conceptos OO puros como: clases, encapsulación, herencia, superclases, subclases, abstracción, polimorfismo, composición, clases abstractas, interfaces java, y otros. Lo cual prueba que la naturaleza del análisis estructural mediante el MEF - elemento viga sistema ejes locales Kardestuncer, es absolutamente orientada a objetos.

El diseño del modelo OO propuesto permite el desarrollo de una aplicación robusta, eficiente y escalable. La implementación se la realizó en Java; sin embargo, el modelo puede programarse en otro lenguaje de objetos como: Smalltalk, C++, Python, y otros.

El caso de estudio probó la factibilidad del modelo OO propuesto en cuanto a concepto y resultados. En cuanto a conceptos porque el modelo mejora las fuerzas internas en los extremos de los elementos conforme Kardestuncer. En cuanto a

MODELO ORIENTADO A OBJETOS PARA ANALIZAR ESTRUCTURAS PLANAS USANDO ELEMENTOS FINITOS

resultados porque las verificaciones manuales basadas en el equilibrio estructural, prueban la exactitud del modelo e implementación.

Finalmente, hay mejoras que realizar en el modelo OO como la inclusión de cargas distribuidas triangulares sobre los elementos, apoyos elásticos, y otros, en virtud de probar la escalabilidad y extender su funcionalidad. Queda realizar más casos de estudio con estructuras más complejas. Y queda comparar los resultados de una aplicación comercial tradicional versus la aplicación resultante del modelo propuesto OO en ésta investigación.

REFERENCIAS

- [1] YAILE CABALLERO "Análisis y Diseño Orientado a Objetos". Postgrado en Informática. Universidad de Camagüey. Cuba. 2007.
- [2] MATT WEISFELD "The Object-Oriented Thought Process". Segunda edición. Editorial Sams. USA. 2004.
- [3] NELSON TERRAZAS "Introducción Práctica a UML". IDELOGIX. LP-BOL. 2007.
- [4] JOSEPH SCHMULLER "Aprendiendo UML en 24 Horas". Editorial Prentice Hall. USA. 2002.
- [5] MARTY HALL "Java 2D Drawing". USA. 2008.
- [6] HORSTMANN Y CORNELL "Core Java 2 Volume II Advanced Features". Editorial Prentice-Hall. USA. 2001.
- [7] CHRISTOS G. BINIARIS "A Three-Dimensional Object-Oriented Distributed Finite Element Solver Based on Mobile Agent Technology". USA. 2004.
- [8] FRANK MCKENNA "A New Design for Object-Oriented Finite Element Software". USA. 2000.
- [9] GRAHAM CHARLES ARCHER "Object-Oriented Finite Element Analysis". University of California at Berkeley. USA. 1996.
- [10] JUN LU "An Object-Oriented Application Framework for Finite Element Analysis in Structural Engineering". USA. 1994.
- [11] KLAUS-JURGEN BATHE "Finite Element Procedures". Editorial Prentice-Hall. USA. 1996.
- [12] ROBERT MELOSH "Structural Engineering Analysis by Finite Elements". Editorial Prentice-Hall. USA. 1990.
- [13] NICHOLAS BARAN "Finite Element on Microcomputers". Editorial McGraw-Hill. USA. 1988.
- [14] MARTIN AND CAREY "Introduction to Finite Element Analysis". Editorial McGraw-Hill. USA. 1973.
- [15] RICHARD GALLAGHER "Finite Element Analysis". Editorial Prentice-Hall. USA. 1975.
- [16] BEER FERDINAND Y JOHNSTON E. RUSSELL "Mecánica Vectorial para Ingenieros". Editorial McGraw-Hill. México. 1993.
- [17] BRAY, CROXTON AND MARTIN "Matrix Analysis of Structures". Editorial Edward Arnold. England. 1976.
- [18] H. KARDESTUNCER "Introducción al Análisis Estructural con Matrices". McGraw-Hill. USA. 1975.
- [19] NORRIS Y WILBUR "Análisis Elemental De Estructuras". Editorial McGraw-Hill. México. 1969.
- [20] PRZEMIENIECKI "Theory of Matrix Structural Analysis". Editorial McGraw-Hill. USA. 1968.
- [21] MOSHE RUBINSTEIN "Matrix Analysis of Structures". Editorial Prentice-Hall. USA. 1966.