

# **SOFTWARE DE SIMULACIÓN CENTRADO EN EL APRENDIZAJE DE LAS CONDICIONES DE EQUILIBRIO**

## *SIMULATION SOFTWARE FOCUSED ON THE LEARNING OF EQUILIBRIUM CONDITIONS*

**Javier Silvestre Zavala**

Tecnológico Nacional de México / Instituto Tecnológico Superior de Irapuato  
*jasilvestre@itesi.edu.mx*

**Juan Manuel Banda Chávez**

Tecnológico Nacional de México / Instituto Tecnológico Superior de Irapuato  
*jmanuelnx@gmail.com*

### **Resumen**

En la actualidad, los estudiantes cuentan con un universo de posibilidades para aprender. Una de ellas es el software de simulación que permite, por ejemplo, observar el desempeño de un proyecto o la solución de un problema, que significan un ahorro significativo de tiempo y esfuerzo. En los tecnológicos del país la Física presenta altos índices de reprobación por diferentes circunstancias. Específicamente en los temas de condiciones de equilibrio, los estudiantes encuentran dificultades para lograr un total entendimiento que se traduzca en la solución de ejercicios y desarrollo de proyectos de aplicación. Es por esto, que el presente trabajo de investigación propone una alternativa de software de simulación que permite resolver problemas de cuerpos en equilibrio en dos dimensiones. Este software presenta una solución paso a paso de los mencionados problemas, incluyendo diagramas de cuerpo libre y material de ayuda sobre los temas.

**Palabras Clave:** Equilibrio, simulación, Software.

### **Abstract**

*Currently, the students have a universe of possibilities to learn. One of them is the simulation software that allows, for example, to observe the performance of a project or the solution of a problem, which mean a significant saving of time and*

*effort. In the technological institutes of the country, Physics presents high failure rates for different circumstances. Specifically in the subjects of equilibrium conditions, students find difficulties to achieve a total understanding that translates into the solution of exercises and development of application projects. It is for this reason that the present research project proposes a simulation software as an alternative that allows solving problems of bodies in equilibrium in two dimensions. This software presents a step-by-step solution of the mentioned problems, including free body diagrams and help material on the topics.*

**Keywords:** Equilibrium, simulation, Software.

## **1. Introducción**

La Física es una ciencia que busca descubrir las causas y efectos de los fenómenos naturales desde el mismo origen del universo. Es por esta razón que es una asignatura esencial en los estudios superiores en diferentes ramas, incluyendo las ingenierías. En este orden de ideas, la solución de problemas fisicomatemáticos puede representar un verdadero reto, en ocasiones insalvable, para un gran porcentaje de estudiantes de ingeniería. La solución implica la comprensión y asimilación adecuada de los temas de la asignatura, lo cual se ve afectado por el nivel de complejidad o los múltiples cálculos requeridos. El efecto de las anteriores complicaciones se ve reflejado en los altos índices de reprobación de la mayoría de los tecnológicos del país. En el Instituto Tecnológico Superior de Irapuato se identificó la dificultad específica de aprendizaje de las condiciones de equilibrio en dos dimensiones. Para coadyuvar a resolver el problema, se desarrolló un software de simulación, ya que la simulación y el aprendizaje son dos conceptos muy unidos en el proceso educativo y, desde el punto de vista puramente instrumental podemos decir que la mayoría de las actividades de aprendizaje siempre están basadas en entidades de simulación [Cabero, 2016]. En la actualidad existen opciones comerciales y libres de software de solución de problemas de Física en general y específicamente de cuerpos en equilibrio. GNU FísicaLab es una aplicación educativa para ordenador para resolver problemas físicos. Su principal objetivo es permitirle al usuario enfocarse en los conceptos

físicos, dejando a un lado los detalles matemáticos [GNU, 2018]. Esta aplicación de software libre contiene los siguientes módulos: Cinemática de partículas en 2D, Cinemática circular de partículas en 2D, Estática de partículas en 2D, Estática de cuerpos rígidos en 2D, Dinámica de partículas en 2D, Dinámica circular de partículas en 2D, Calorimetría, gases ideales y dilatación. Esta aplicación solo presenta los resultados de los problemas.

Interactive Physics es un software de ordenador que permite modelar, simular y explorar una amplia variedad de fenómenos físicos y crear una gran cantidad de experimentos [Interactive Physics, 2018]. Este software permite entre otras cosas: crear objetos dibujando círculos, bloques y polígonos; simular contacto, colisiones y fricción; ver resultados como números, gráficas y vectores animados; medir velocidad, aceleración, fuerza, energía, etc. Por su parte, Staticus2d es una aplicación para dispositivo móvil que permite hallar dos fuerzas desconocidas P y Q cuando un cuerpo sometido a fuerzas concurrentes está en equilibrio. Otra aplicación para móviles es Dinámica del plano inclinado, la cual calcula numéricamente y representa vectorialmente las principales variables que intervienen en los problemas de Dinámica del plano inclinado. Esta aplicación ofrece una pantalla con la resolución detallada, los cálculos intermedios y la representación vectorial de todas las fuerzas que intervienen en el problema.

La solución presentada en este trabajo de investigación de un software de simulación para solución de problemas de condiciones de equilibrio ofrece ventajas puntuales sobre los softwares mencionados líneas arriba. Las ventajas estriban principalmente en que además de arrojar el resultado, ofrece una solución paso a paso de los problemas más comunes de cuerpos en equilibrio traslacional y rotacional y no solo un caso específico. Se ofrece el diagrama de cuerpo libre con todas las fuerzas que actúan y su descomposición en componentes. Lo anterior permite que el estudiante utilice el simulador para incentivar un aprendizaje significativo que lo conduzca a la creación de estructuras de conocimiento mediante la relación sustantiva entre la nueva información y sus ideas previas [Díaz, 2002].

## **2. Métodos**

La investigación se basó en una combinación de los modelos cuantitativo y cualitativo conocida como paradigma mixto. El enfoque mixto implica un proceso de recolección, análisis y vinculación de datos cuantitativos y cualitativos en un mismo estudio o una serie de investigaciones para responder al planteamiento del problema [Hernández, 2006].

Los hechos cuantificables en la investigación fueron los exámenes para medir el impacto del software de simulación para el aprendizaje de las condiciones de equilibrio. También, se recopiló información cualitativa como el contexto en que se desarrollan las cátedras, las experiencias previas de los estudiantes en otras materias, y su actitud frente a las cosas o conocimientos en base al significado que tiene para ellos.

Las variables presentes en el trabajo de investigación son:

- **Variable independiente.** Software de simulación centrado en el aprendizaje de las condiciones de equilibrio.
- **Variable dependiente.** Aprendizaje del estudiante.

### **Ciclo de vida**

Para el desarrollo del software se utilizó el ciclo de vida iterativo, ya que este ciclo conlleva el refinamiento sucesivo de una arquitectura orientada a objetos por el cual se aplica la experiencia y resultados de cada versión a la siguiente iteración del análisis y diseño [Booch, 1998]. En la figura 1 se muestra el ciclo de vida iterativo.

### **Identificación de la necesidad del software de simulación**

Para identificar la necesidad del software de simulación, se aplicó un cuestionario de preguntas cerradas a estudiantes de tercer semestre de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior de Irapuato. Por otro lado, se obtuvieron estadísticas sobre índices de reprobación de la materia de Física en temas de condiciones de equilibrio desde el 2012 y hasta la fecha de realización de esta investigación en el mencionado instituto y carrera.

Los resultados de las preguntas más relevantes del cuestionario y los índices de reprobación se observan en la figura 2.

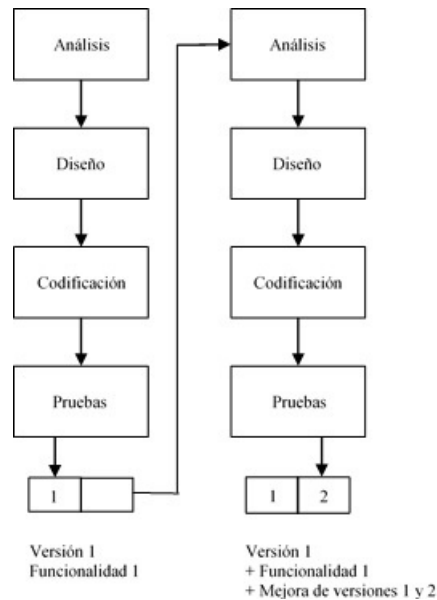


Figura 1 Ciclo de vida.

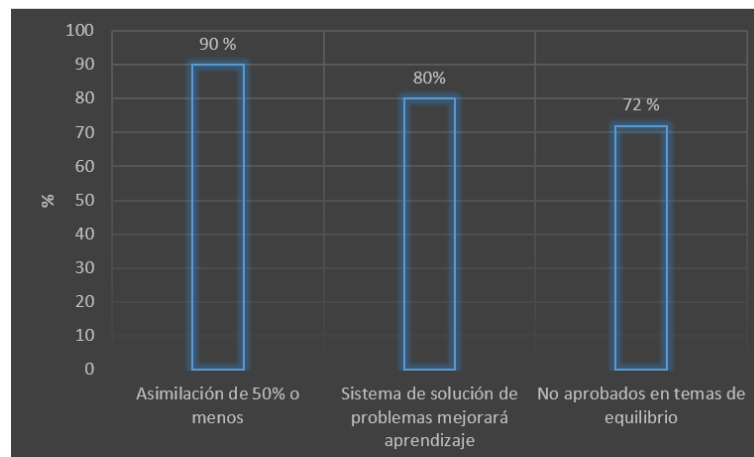


Figura 2 Identificación de las necesidades del sistema.

Estos resultados muestran que el 90% de los estudiantes a quienes se aplicó el cuestionario, refirieron que asimilaban 50% o menos de los temas de cuerpos en equilibrio. A pregunta expresa sobre si consideraban que un software de simulación para resolver ejercicios de cuerpos en equilibrio ayudaría a mejorar su aprendizaje sobre los temas, el 80% contestó de forma afirmativa. Por último, se

muestra también en la figura 2 que el 72% de los estudiantes reprobaron los temas de condiciones de equilibrio.

Por la información presentada en esta sección es que se decidió desarrollar el software de simulación centrado en el aprendizaje de las condiciones de equilibrio.

### **Requerimientos de desarrollo del simulador**

Para el desarrollo del simulador se llevó a cabo un análisis de requerimientos, el cual da como resultado la especificación de las características operativas del software, indica la interfaz y otros elementos del sistema [Pressman, 2010]. Dentro de los requerimientos funcionales se estableció que el programa debía realizar los cálculos de los problemas más comunes de cuerpos en equilibrio, mostrando los resultados paso a paso junto con los diagramas de cuerpo libre con descomposición vectorial para que el usuario pudiera tener una mejor comprensión del problema. Respecto a los requerimientos no funcionales se tuvo que el simulador sería desarrollado en el lenguaje de programación C# en el Framework 4.5 con la IDE "Visual Studio". Para que el software pueda correr, el equipo de cómputo debe tener las siguientes características mínimas: procesador 1.4 GHz, 512 MB RAM, 1 GB de espacio en disco duro.

### **Casos de uso**

Respecto a los casos de uso, los cuales narran una historia estilizada sobre como interactúa un usuario final (que tiene cierto número de roles posible) con el sistema en circunstancias específicas [Pressman, 2010], se consideraron cuatro: iniciar software, seleccionar problema, insertar datos del problema y cerrar software. En la tabla 1 se muestran los eventos del caso de uso de insertar datos del problema.

### **Elementos y funciones para el desarrollo**

Para el desarrollo del simulador se utilizó el IDE Visual Studio, el cual es el Entorno de Desarrollo Integrado (IDE) de Microsoft para crear, ejecutar y depurar programas escritos en una variedad de lenguajes de programación .NET [Deitel,

Deitel, 2007]. Se utilizaron diferentes librerías y ensamblados externos al lenguaje C# los cuales se describen a continuación:

- **System.Drawing.Graphics.** La clase Graphics proporciona métodos para dibujar objetos en la pantalla. Se pueden dibujar muchas formas y líneas diferentes utilizando un objeto Graphics. Estos métodos incluyen DrawLine, DrawArc, DrawClosedCurve, DrawPolygon, y DrawRectangle. También puede dibujar imágenes e iconos utilizando la función DrawImage y DrawIcon, respectivamente. Además, puede manipular el sistema de coordenadas.
- **Ensamblado de Microsoft.VisualBasic.** Se utilizó esta librería ya que para el desarrollo del proyecto se necesitaron controles que C# no tiene por defecto.

Tabla 1 Insertar datos del problema.

Caso de uso:		Insertar datos del problema	
Actores:		Usuario, Alumno, Maestro.	
Descripción:		Insertar los datos para resolver un problema en específico.	
Precondición:		Un problema de condición de equilibrio debe estar seleccionado.	
Curso Normal			
Eventos usuarios		Eventos Sistema	
		1	El software muestra la interfaz del problema seleccionado.
2	El usuario inserta los datos en la sección de entradas.		
3	El usuario presiona el botón de resolver.		
		4	El software muestra el procedimiento de solución paso a paso en el apartado de resultados.
Cursos alternos			
		5	Si los datos son ingresados incorrectos el programa muestra un mensaje de error.
		5.1	Si los datos ingresados generan desbordamiento, esa parte del programa se cerrará automáticamente.
Postcondición		El sistema está listo para seleccionar un nuevo tipo de problema o salir.	

## Desarrollo del software de simulación

Debido a que los usuarios comprenden una aplicación al visualizar su interfaz gráfica de usuario (GUI) [Braude, 2003], en el software de simulación la interfaz para todos los casos de cuerpos en equilibrio está formada del mismo número de controles. Dicha interfaz tiene dos secciones de dibujo, una de ellas muestra la figura generada al ingresar los datos, dibujando sobre un sistema de ejes coordenados los vectores ingresados, mientras que la otra muestra el diagrama de cuerpo libre con la correspondiente descomposición de vectores. Contiene también una sección donde el usuario ingresa los datos del problema y finalmente una que muestra el desarrollo del problema paso a paso. Lo anterior se puede apreciar en la figura 3.

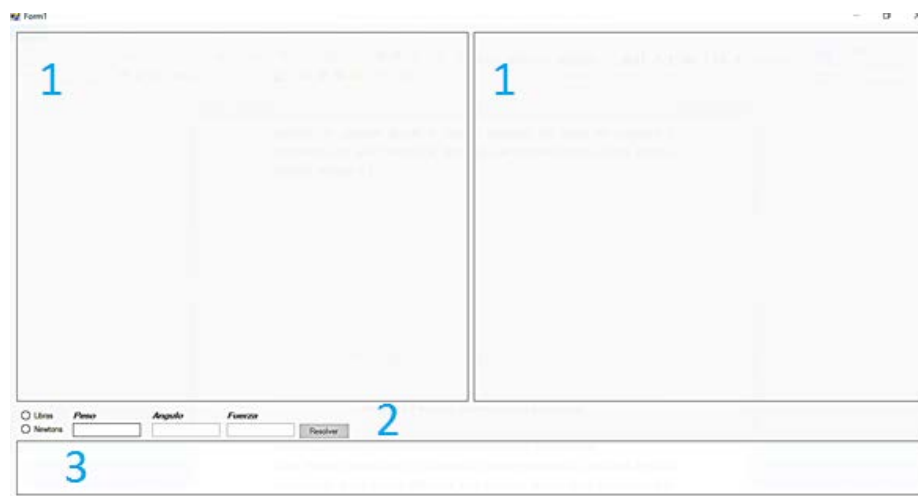


Figura 3 Interfaz general del simulador.

Para desarrollar el proyecto se utilizaron 2 clases, las cuales describen un grupo de objetos con estructura y comportamiento común [Weitzenfeld, 2005]. Las clases fueron llamadas “Dibujo” y “Cálculo”. La clase Dibujo se encarga de trazar el gráfico que se forma a partir de los datos de entrada, así como el diagrama de cuerpo libre. En la figura 4 se muestra parte del código de esta clase.

En la clase Cálculo se realizan los diferentes cálculos para llegar a la solución, mostrando el proceso paso a paso en el control de resultados. En la figura 5 se muestra parte del código de esta clase.



```
miDibujo.DrawLine[cuadro, 0, i, miMapa.Width, i];
}

Random rdn = new Random();

for (int i = 0; i < contDatos; i++)
{
    double Angulo = (DatosCos[i, 0] * Math.PI) / 180 * -1; //Radianes
    //double py = ((radio) * Math.Sin(Angulo en radianes) + (origen));

    double px = ((this.Objpic.Width) * Math.Cos(Angulo) + (this.Objpic.Width / 2));
    double py = ((this.Objpic.Height) * Math.Sin(Angulo) + (this.Objpic.Height / 2));

    int r = rdn.Next(1, 1);
    int g = rdn.Next(100, 200);
    int b = rdn.Next(150, 230);

    Pen lapiz = new Pen(Color.FromArgb(r, g, b), 1);
    /*lapiz.StartCap = LineCap.ArrowAnchor; Fechas ->
    lapiz.EndCap = LineCap.ArrowAnchor;*/
    miDibujo.DrawLine(lapiz, this.Objpic.Width / 2, this.Objpic.Height / 2, (int)px, (int)py);
    Brush letra = new SolidBrush(Color.FromArgb(0, 0, 0));
    px = ((this.Objpic.Width / 4) * Math.Cos(Angulo) + (this.Objpic.Width / 2));
    py = ((this.Objpic.Height / 4) * Math.Sin(Angulo) + (this.Objpic.Height / 2));
    miDibujo.DrawString(DatosCos[i, 0].ToString() + "°", myFont, letra, (int)px, (int)py);
}

for (int i = 0; i < 2; i++)
{
    double Angulo = (Variable[i] * Math.PI) / 180 * -1; //Radianes
```

Figura 4 Parte del código de la clase dibujo.

```
{
    signo[i, 0] = "+"; //Coseno
    signo[i, 1] = "+"; //Seno
    ecuacion1 += signo[i, 0] + nomvar[i] + Unidad + "Cos " + Variable[i];
    ecuacion2 += signo[i, 1] + nomvar[i] + Unidad + "Sen " + Variable[i];
}

if (Variable[i] > 90 && Variable[i] <= 180)
{
    Variable[i] = 180 - Variable[i];

    signo[i, 0] = "-"; //Coseno
    signo[i, 1] = "+"; //Seno
    ecuacion1 += signo[i, 0] + nomvar[i] + Unidad + "Cos " + Variable[i];
    ecuacion2 += signo[i, 1] + nomvar[i] + Unidad + "Sen " + Variable[i];
}

else if (Variable[i] > 180 && Variable[i] <= 270)
{
    Variable[i] = 270 - Variable[i];
    signo[i, 0] = "-"; //Coseno
    signo[i, 1] = "-"; //Seno
    ecuacion1 += signo[i, 0] + nomvar[i] + Unidad + "Cos " + Variable[i];
    ecuacion2 += signo[i, 1] + nomvar[i] + Unidad + "Sen " + Variable[i];
}

else if (Variable[i] > 270 && Variable[i] <= 360)
{
    Variable[i] = 360 - Variable[i];
    signo[i, 0] = "+"; //Coseno
    signo[i, 1] = "-"; //Seno
    ecuacion1 += signo[i, 0] + nomvar[i] + Unidad + "Cos " + Variable[i];
```

Figura 5 Parte del código de la clase Cálculo.

### 3. Resultados

Una vez terminado el software de simulación centrado en el aprendizaje de las condiciones de equilibrio, se procedió a probar su eficacia en dos grupos de la

materia de Física de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior de Irapuato. Se aplicaron los exámenes escritos correspondientes a los temas de cuerpos en equilibrio del semestre. Posteriormente, se facilitó el software a los estudiantes para que lo utilizaran y se aplicaron nuevos exámenes escritos. Los resultados de ambas evaluaciones, la preliminar y la posterior al uso del sistema, se muestran en la figura 6. Se puede observar una mejora significativa en los índices de aprobación relacionados directamente con la variable dependiente de la investigación: el aprendizaje de los estudiantes. Los exámenes escritos aplicados sin utilizar el simulador arrojaron un 78% de reprobación, mientras que después de utilizar el software de simulación se obtuvo un porcentaje de reprobación de 38%.

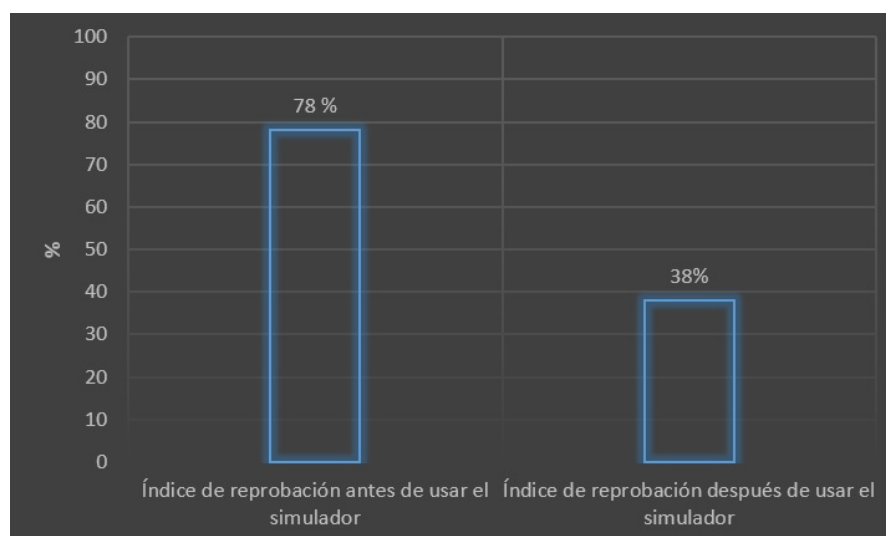


Figura 6 Resultados del examen escrito antes y después del uso del sistema.

Por otro lado, se aplicó una encuesta de satisfacción a los estudiantes después de utilizar el software de simulación centrado en el aprendizaje de las condiciones de equilibrio, cuyos resultados se muestran en la figura 7.

Los números arrojados por el instrumento muestran que la satisfacción en términos generales fue bastante alta, ya que el 98% de los encuestados consideró que el software les ayudó en el aprendizaje de las condiciones de equilibrio y, para quienes lo lograron, en la aprobación de la evaluación respectiva. Además, el

software satisfizo a los estudiantes en los apartados de diseño de la interfaz gráfica y usabilidad como lo muestran los resultados del 90% y 95% respectivamente. El 80% consideró que cada caso de las condiciones de equilibrio presentado en el software estuvo suficientemente descrito.

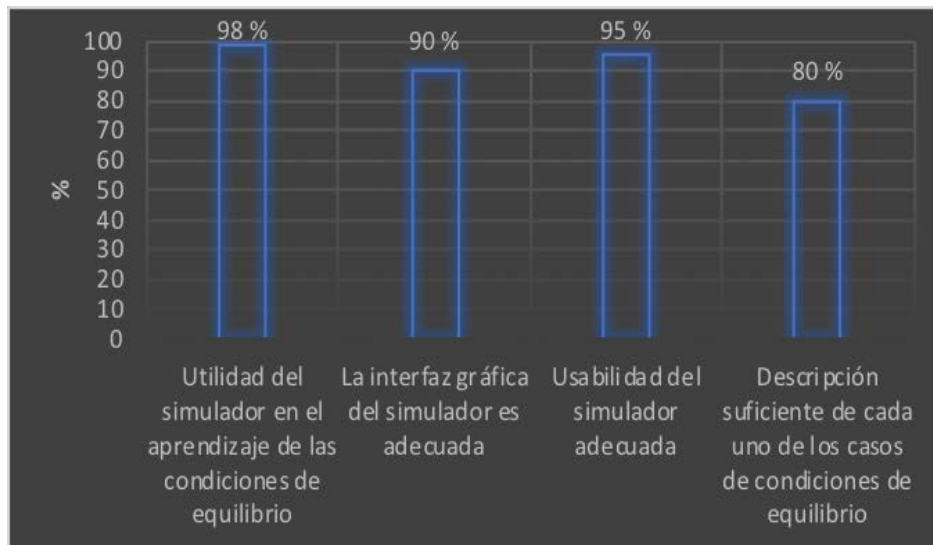


Figura 7 Resultados de la encuesta de satisfacción después del uso del sistema.

En la encuesta se incluyó una pregunta acerca de las mejoras que los estudiantes sugerían al software. Las respuestas más recurrentes indicaron que el software contara con una sección que presentara información teórica sobre las condiciones de equilibrio, una sección de autoevaluación y que se cubrieran las condiciones de equilibrio en tres dimensiones. Lo anterior dio pie a la identificación de mejoras a introducir en versiones futuras del software.

### Software de simulación centrado en el aprendizaje de las condiciones de equilibrio

La interfaz del software de simulación cuenta de inicio con una pantalla que muestra los 10 casos diferentes de problemas de cuerpos en equilibrio que se pueden resolver, como se muestra en la figura 8. Cabe mencionar que los primeros 7 se refieren a equilibrio traslacional, mientras que los últimos tres se refieren a equilibrio traslacional y rotacional.

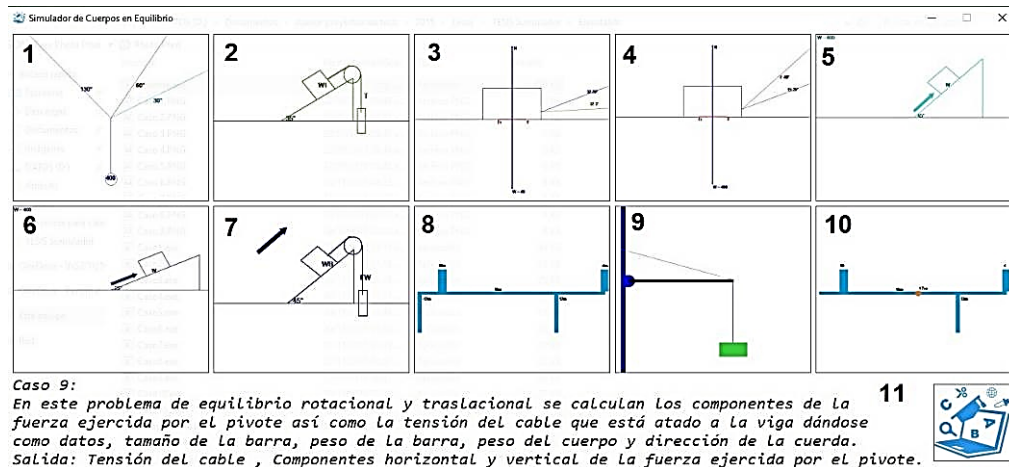


Figura 8 Interfaz del simulador para solución de cuerpos en equilibrio.

En el caso 1 se resuelven las tensiones de las cuerdas que sostienen un cuerpo. El caso 2 calcula la fuerza paralela al plano inclinado debida a la interacción de dos cuerpos. El caso 3 calcula los coeficientes de fricción cinética y estática por el movimiento de un cuerpo. El caso 4 calcula la fuerza que se requiere para arrastrar un cuerpo. El caso 5 calcula la fuerza paralela al plano para que un cuerpo se mueva hacia arriba o hacia abajo con velocidad constante. El caso 6 calcula si un cuerpo está en equilibrio o no. El caso 7 calcula la fuerza de fricción en un plano inclinado y la dirección del movimiento. El caso 8 calcula las fuerzas de reacción ejercidas por los soportes sobre una barra. El caso 9 calcula las componentes de una fuerza ejercida por un pivote y la tensión en una cuerda. El caso 10 calcula el momento de torsión resultante respecto a un punto. La sección 11 de la interfaz muestra una breve descripción de lo que se calcula en cada caso, así como los datos que se ingresan y los resultados arrojados. En la figura 8 se observa que se encuentra seleccionado el caso 9. Para mostrar el contenido de la sección 11 basta con colocar el puntero del mouse sobre el caso correspondiente.

A continuación, se detalla el funcionamiento del simulador tomando un problema del caso 5, el cual se selecciona haciendo clic con el mouse sobre su sección en la pantalla principal. En la figura 9 se muestra como después de ingresar los datos se crea el diagrama de cuerpo libre con la descomposición vectorial en componentes cuando el cuerpo sube.

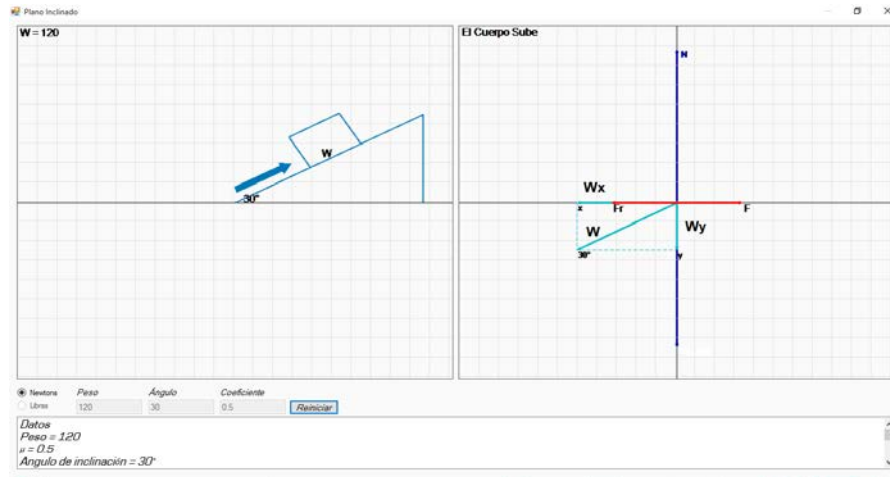


Figura 9 Diagrama de cuerpo libre cuando el cuerpo sube.

Para mostrar el diagrama de cuerpo libre cuando el cuerpo baja como se observa en la figura 10, basta con hacer doble clic en la sección de diagrama de cuerpo libre de la interfaz. Es importante hacer notar que en la sección de datos podemos elegir entre Newtons o Libras para el peso, y que una vez resuelto el problema podemos reiniciar con otro problema diferente.

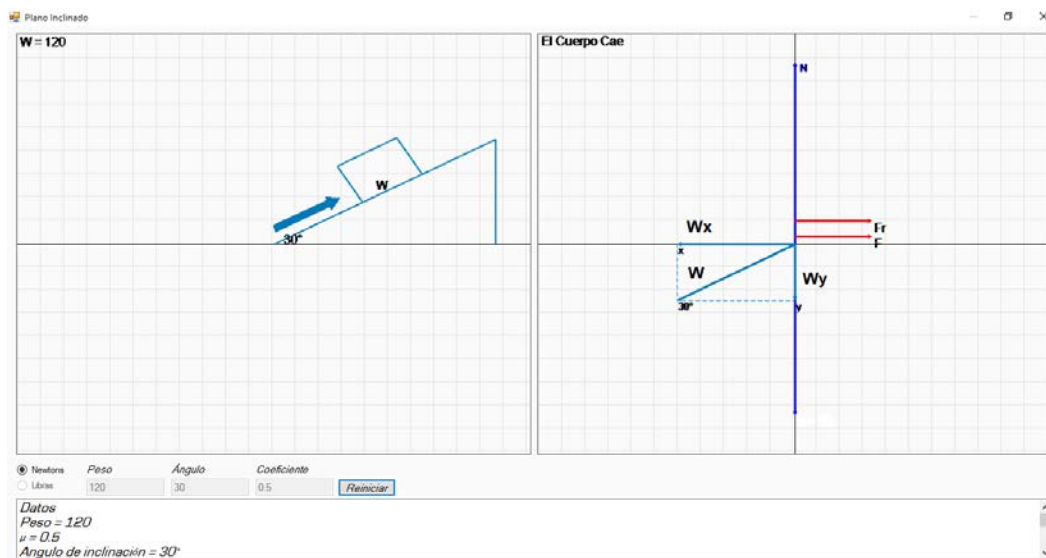


Figura 10 Diagrama de cuerpo libre cuando el cuerpo cae.

Por último, para observar el proceso de solución paso a paso se procede a un desplazamiento con la barra correspondiente en el panel de resultados, o bien, se

puede hacer doble clic en el mencionado panel para ampliar la pantalla, como se observa en la figura 11. Para regresar a la vista anterior basta hacer doble clic en la pantalla extendida.

```

Piano Inclinado
Datos
Peso = 120
μ = 0.5
Angulo de inclinación = 30°
Nota
μs = Coeficiente de rozamiento estatico
μk = Coeficiente de rozamiento dinamico
Fr = Fuerza de rozamiento
Frs = μsN
Frk = μkN
∑Fy = 0
N - Wcosθ = 0
N - 120Ncos30 = 0
N + [-103.923048454133N] = 0
N = 103.923048454133N
Fuerza con velocidad hacia arriba
∑Fx = 0
F - Fr - Wx = 0
Fk = μkN
F - μkN - Wx = 0
F = μkN + Wx
F = 0.5N(103.923048454133N) + 120Nsen30
F = 111.961524227066N

Fuerza con velocidad hacia abajo
∑Fx = 0
F + Fr - Wx = 0
F = Wx - Fr
F = 120Nsen30 + [-0.5N](103.923048454133N)
F = 0.03847577293367N
    
```

Figura 11 Solución paso a paso pantalla extendida.

### Manual de usuario

El software de simulación se acompaña de un manual de usuario, el cual incluye las instrucciones para el manejo del software. En la figura 12 se muestra un extracto de dicho manual donde se explica cómo ingresar los datos cuando se selecciona un problema correspondiente al caso 1 donde se calculan las tensiones de las cuerdas que sostienen a un cuerpo.

Como existen dos incógnitas en este ejemplo se inserta el valor de cada una, y en el campo "Fuerza" se inserta T1 y T2, insertamos el valor presionando enter en el campo de la fuerza.

Ángulo	Fuerza
<input type="text" value="20"/>	<input type="text" value="t1"/>

Ángulo	Fuerza
<input type="text" value="120"/>	<input type="text" value="t2"/>

Si existiese una tercera cuerda se tendría que insertar forzosamente el valor de la fuerza aplicada a la cuerda. Ejemplo:

Ángulo	Fuerza
<input type="text" value="50"/>	<input type="text" value="23"/>

Figura 12 Extracto del manual de usuario.

## **4. Discusión**

Las ciencias duras tales como la Física, han significado un dolor de cabeza para los diferentes actores que participan en el proceso de enseñanza aprendizaje, díganse el estudiante, el profesor y las instituciones educativas. Para el estudiante significa atrasarse en sus planes de estudio al no aprobar la materia, y en algunos casos extremos es causa de su deserción de la institución. El profesor, por su parte, se enfrenta a la búsqueda incesante de nuevas estrategias que le permitan impactar eficazmente en el aprendizaje de los estudiantes, buscando subsanar las deficiencias que arrastra de niveles de estudio anteriores. Las instituciones, por otro lado, buscan aumentar sus índices de egreso, egresados que lleven una sólida preparación que les permita enfrentarse al mercado laboral de forma competitiva. En este orden de ideas, el uso de herramientas de software que faciliten el aprendizaje de los estudiantes de ingeniería complementa efectivamente lo aprendido en las asignaturas del plan de estudio que cursan. Si el estudiante tiene a su disposición los medios que le permitan no solo comprobar resultados, sino que le muestren el proceso completo de solución de los problemas, entonces contará con uno de los elementos que lo llevan a aprobar las materias no solo por productos sino, y, sobre todo, por un aprendizaje significativo.

Por otro lado, la encuesta de satisfacción que se aplicó a los estudiantes en este trabajo de investigación arrojó como áreas de oportunidad del software la ausencia de información teórica sobre los temas que trata y una sección de autoevaluación. De tal manera que un software de simulación que incluya estos elementos, además de la solución paso a paso, puede convertirse no solo en una herramienta que complemente el aprendizaje que los estudiantes adquieren en las aulas, sino en una que permita el autoaprendizaje sin la necesidad imperiosa de un profesor.

## **5. Conclusiones**

En el presente trabajo se generó un software de simulación centrado en el aprendizaje de las condiciones de equilibrio. Este simulador permite resolver diez casos diferentes de problemas que implican equilibrio traslacional y rotacional. En el caso 1 se resuelven las tensiones de las cuerdas que sostienen un cuerpo. El

caso 2 calcula la fuerza paralela al plano inclinado debida a la interacción de dos cuerpos. El caso 3 calcula los coeficientes de fricción cinética y estática por el movimiento de un cuerpo. El caso 4 calcula a fuerza que se requiere para arrastrar un cuerpo. El caso 5 calcula la fuerza paralela al plano para que un cuerpo se mueva hacia arriba o hacia abajo con velocidad constante. El caso 6 calcula si un cuerpo está en equilibrio o no. El caso 7 calcula la fuerza de fricción en un plano inclinado y la dirección del movimiento. El caso 8 calcula las fuerzas de reacción ejercidas por los soportes sobre una barra. El caso 9 calcula las componentes de una fuerza ejercida por un pivote y la tensión en una cuerda. El caso 10 calcula el momento de torsión resultante respecto a un punto.

El simulador permite observar la solución completa de los problemas paso a paso, mostrando los diagramas de cuerpo libre con la descomposición vectorial en componentes. Además, se incluye un manual de usuario que explica detalladamente el funcionamiento del simulador.

Con el presente trabajo se pretende que los estudiantes de Física de las carreras de Ingeniería impartida en los Institutos Tecnológicos del país, y en general cualquier estudiante interesado en el tema, puedan complementar su aprendizaje sobre condiciones de equilibrio. Se pretende que lo anterior incida positivamente en los índices de aprobación por aprendizaje significativo de la materia de Física. Por otro lado, se prevé a futuro complementar el trabajo integrando la solución de problemas de cuerpos en equilibrio en tres dimensiones, información teórica sobre condiciones de equilibrio y una sección de autoevaluación. Lo anterior permitirá una mayor versatilidad con fines didácticos, ya que el estudiante contará con una sola herramienta que le ofrezca la ayuda completa en los temas de equilibrio de la materia de Física.

Por último, los resultados obtenidos al utilizar el simulador muestran una mejoría significativa en la asimilación, comprensión y aplicación de conceptos relacionados con condiciones de equilibrio en la solución de problemas. De tal manera que se comprobó la utilidad de las herramientas didácticas de software en el aprendizaje de los estudiantes de ingeniería.



## **6. Bibliografía y Referencias**

- [1] Booch G., Análisis y Diseño Orientado a Objetos con Aplicaciones, Pearson Educación, Segunda Edición, 1998, 265, 266 pp.
- [2] Braude E., Ingeniería de Software, una perspectiva orientada a objetos, Alfaomega, 2003, 151 p.
- [3] Cabero-Almenara, J., Costas J., La utilización de simuladores para la formación de los alumnos. Prisma Social 2016: <<http://ucsj.redalyc.org/articulo.oa?id=353749552015>> ISSN, 6 p.
- [4] Deitel H., Deitel P, Como programar C#, Pearson, Segunda Edición, 2007, 18 p.
- [5] Díaz F., Hernández G., Estrategias Docentes para un Aprendizaje Significativo. Una Interpretación Constructivista, Mc Graw Hill, Segunda Edición, 2002, 39 p.
- [6] Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. Metodología de la investigación, Mc-Graw Hill, 2006, 783 p.
- [7] Pressman R., Ingeniería del Software Un Enfoque Práctico, Mc Graw Hill, Séptima Edición, 2010, 127 p.
- [8] Weitzenfeld A., Ingeniería de Software Orientada a Objetos con uml, java e internet, Thomson, 2005, 72 p.
- [9] <https://www.gnu.org/software/fisicalab/index.es.html> junio de 2017.
- [10] <http://www.design-simulation.com/IP/Index.php> junio de 2017.