Simulación de una Fuerza Central usando EJS para una clase de Física

Rubén Sánchez Sánchez
Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada del Instituto Politécnico Nacional. Unidad Legaria. Calzada Legaria No. 694. Colonia Irrigación. Delegación: Miguel Hidalgo. C.P. 11500, Ciudad de México.

E-mail: rsanchezs@ipn.mx

(Recibido el 13 de Julio de 2016, aceptado el 25 de Agosto de 2016)

Resumen

Este documento contiene sugerencias académicas para llevar a cabo una clase de Física, utilizando para ello una TIC consistente en un programa de diseño de simulaciones físicas. El paquete fue diseñado por Esquembre de la Universi dad de Murcia en España, y representa una solución gratuita para implementarla en el salón de clases. En este caso vamos a preparar una simulación de una partícula sujeta a un campo de fuerza central. La fuerza de ligamiento obedece la ley de los resortes. Así que la simulación tiene una formulación matemñatica sencilla que pueden entender los estudiantes de una carrera de ingeniería o ciencias. Se recomienda implementarla junto a una estrategia didáctica, convirtiendo la clase en una Clase Demostrativa Interactiva, a la manera en como lo sugieren Sokoloff *et al.* Esperamos que este trabajo sea de utilidad a los profesores que den la materia de Física para carrerars de ingeniería o ciencias exactas. Y que los estudiantes aprovechen los beneficios de la TIC y las Clases Demostrativas Interactivas.

Palabras clave: Fuerza central, ley de Hooke, simulaciones, enseñanza de la Física.

Abstract

This document contains academic suggestions for conducting a Physics class, using a TIC consisting of a physical simulations design program. The package was designed by Esquembre of the University of Murcia in Spain, and represents a free solution to implement it in the classroom. In this case we are going to prepare a simulation of a particle subject to a central force field. The bonding force obeys the law of the springs. So the simulation has a simple mathematical formulation that students of an engineering or science career can understand. It is recommended to implement it together with a learning strategy, by turning the class into an Interactive Demonstrative Class, as suggested by Sokoloff *et al*. We hope that this work will be useful to teachers who give the physics subject to engineering or exact sciences. And that the students take advantage of the benefits of ICT and the Interactive Demonstrative Classes

Keywords: Central force, Hooke's law, simulations, Physics education.

PACS: 02.30.Hq, 01.40.-d, 01.50.-i, 01.50.H- **ISSN 1870-9095**

I. INTRODUCCIÓN

No es de extrañar que en los últimos años las Tecnologías de la Información y la Comunicación han alcanzado su lugar dentro del ámbito educativo, como material auxiliar para impartir clases. Debido a su versatilidad y la facilidad con que ahora disponemos de computadoras en la escuela y de computadoras personales, no resulta una mala idea tratar de implementar una clase de Física utilizando como auxiliar tecnológico a una computadora.

La computadora resulta ser un medio que para el profesor ofrece facilidad y economía.

Para los estudiantes promedio, un computador puede significar innovación en la forma de aprender una clase o lección de Física.

En este trabajo veremos como utilizar la herramienta de software diseñada por Esquembre [1, 2], para preparar simulaciones de fenómenos físicos. En particular veremos como preparar la simulación de una partícula moviéndose en un campo de fuerza central que es proporcional a la distancia de la partícula al centro. Esto es, la fuerza que mantiene ligada a la partícula al campo sigue la ley de Hooke de los resortes.

Dada la facilidad de la herramienta, entonces nos hemos puesto a describir, en este trabajo como se puede editar tal simulación, para que el profesor de Física tenga una buena alternativa de trabajo y actividades recreativas para el estudiante, aparte de el aprendizaje teórico del mismo.

Esto hace que la clase de Física sea más interesante y promueve la actitud activa de los estudiantes en lugar de la tradicional postura pasiva.

Además el estudiante es estimulado a aprender un método numérico de aproximación para resolver ecuaciones diferenciales, el método recomendado, tanto por su sencillez como por su precisión es el método numérico del punto medio, que es un método de segundo orden, y es bastante bueno para aproximar la evolución de un cuerpo bajo la fuerza central de resorte.

Esperamos que este trabajo contribuya en el quehacer diario del profesor de Física y le abra buenas alternativas de ayuda en su enseñar y de actividades didácticas alternativas.

II. USO DE EASY JAVA SIMULATIONS

Easy Java Simulations o EJS para abreviar es un software didáctico para escribir programas que simulan varios

fenómenos Físicos. La razón de recomendarlo para una clase de Física, es que el material es de fácil adquisición en la web, y es de licencia libre. EJS fue desarrollado por Francisco Esquembre de la Universidad de Murcia en España, y debido a su versatilidad es altamente recomendado como un software de desarrollo de simulaciones de tipo *open source* o de código abierto. La página actual del software es [2]:

http://fem.um.es/Ejs/

En las siguientes secciones veremos como esta herramienta es fácil de usar y puede mejorar la calidad de una clase de Física. Esta puede ser a nivel Universitario o Preparatorio.

III. DESCRIPCIÓN

Para desarrollar cualquier simulación en EJS es necesario seguir tres fases de edición, que son

- Descripción
- Modelo
- Vista

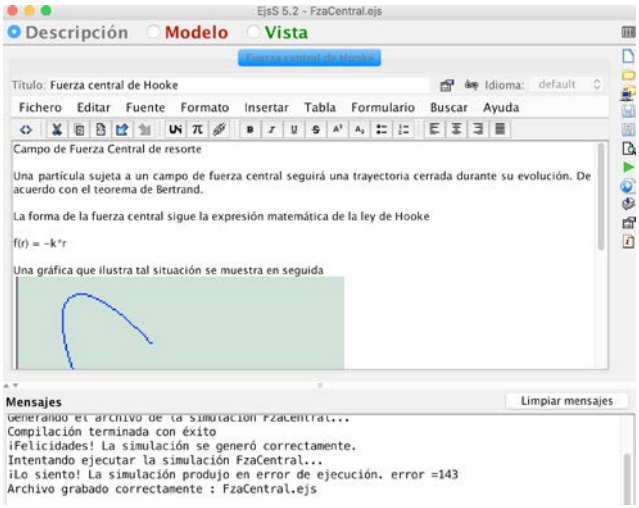

FIGURA 1. Descripción del fenómeno físico.

Al bajar EJS de su sitio y desempaquetarlo, nos deja con una carpeta de nombre más o menos EjsS—versión--, donde –versión representa la versión del programa EJS que hemos bajado. Lo importante es que dentro de la carpeta hay un ejecutable en formato j*ar* de Java, llamado generalmente EjsConsole.jar. El archivo es autoejecutable, y si hemos instalado Java en nuestro computador, basta con darle un click izquierdo para que el programa abra y empiece a funcionar. Después de darla algunos datos personales y de decidir en que carpeta se va a trabajar, EJS muestra dos ventanas, que son:

- Consola de Ejs
- Ejs

Como su nombre lo menciona la primera es la consola de Ejs, donde se pueden manejar atributos generales del programa como por ejemplo, donde va a ser nuestro espacio de trabajo, entre otros aspectos generales de Java. La segunda ventana varía en realidad de nombre específico pero generalmente empieza su nombre como lo enlistamos. Representa la ventana principal del programa que sirve para editar el programa, y es esta ventana donde siempre habremos de trabajar para editar nuevas simulaciones. En la parte superior de esta ventana existe tres radio botones que representan las tres fases de edición del nuevo programa simulador, estas fases son tres y son las siguientes

- Descripción
- Modelo
- Vista

En la Descripción hallamos un pequeño editor de texto, al tipo Word o Pages pero mucho más sencillo. Como aquí lo importante es recabar información del fenómeno físico que se quiera simular, es sencillo de editar. Además el sencillo editor permite incluir gráficas que ayuden a describir y precisar mejor al fenómeno. Sin precisar más detalles ponemos un ejemplo de esta ventana en la figura 1.

En la siguiente sección abordaremos el problema de editar el Modelo de la simulación. Aquí es donde, el estudiante o el profesor puede escribir las ecuaciones de movimiento del fenómeno (en este caso de u cuerpo o partícula sujeta a un campo de fuerza central, donde la fuerza sigue la ley de Hooke). Y puede hacer un desarrollo de evolución del sistema utilizando estas mismas ecuaciones. Como lo hemos dicho antes, aquí utilizaremos el método numérico del punto medio. Para hallar la teoría del método recomendamos varias referencias [3] de la literatura.

IV. VARIABLES

Si elegimos el segundo radio botón del menú superior de EJS, tendremos la ventana de edición para escribir el Modelo de la simulación. EJS ofrece la alternativa de escribir sólo la ecuación diferencial y resolverla con uno de sus métodos numéricos integrados. Como aquí estamos interesados en resolver el problema de acuerdo al método del punto medio, escrito por nosotros, entonces elegimos esta opción y nos disponemos a editar el desarrollo en el movimiento de la partícula sujeta a un campo de fuerza central. Para encontrar la teoría, desarrollada de manera sutil y clara del campo de fuerza central, se puede consultar la obra de Herbert Goldstein [3]. Aquí retomamos sus ecuaciones con un campo de fuerza que sigue la ley de Hooke. Esto es, la fuerza central es proporcional a la distancia *r* de la partícula al campo de fuerzas

$$f(r) = -k\,r. \qquad (1)$$

Para editar el Modelo hay seis etapas o pasos que hay que editar, y que están representados por seis radio botones situados en una línea inferior a la de donde esta el radio botón Modelo, estos son: Variables, Inicialización, Evolución, Relaciones fijas, Propio, Elementos. Pero aquí sólo necesitamos tres pasos que son

- Variables
- Evolución
- Relaciones fijas

En la parte de Variables mencionamos a las variables relevantes que participan en la evolución del movimiento. También está parte donde editamos la evolución del

movimiento y tratamos numéricamente la solución de la ecuación de movimiento. Y finalmente esta la etapa de las Relaciones fijas, donde se anotan las ecuaciones o relaciones matemáticas entre variables, que se mantendrán sin cambio, durante todo el tiempo que dure la simulación.

En la tabla I, mencionamos las variables y sus valores iniciales, hay que tomar en cuenta que todas las variables son de tipo doublé, esto es, son de aritmética de punto flotante de doble precisión dentro del computador.

Vamos a tratar de describir estas variables brevemente: La variable *t* representa el flujo del tiempo desde que inicia la simulación, durante la simulación y hasta que ésta termina. Se le asigna en forma natural el valor inicial de cero *0 seg*, y se va incrementando poco a poco mediante un incremento de tiempo llamado *h=dt*. Luego está la masa de la partícula en movimiento que designamos con la letra m. En la simulación podemos utilizar coordenadas polares, para localizar la posición de la partícula, para esto están las variables *r* y theta. Donde r es la distancia de la partícula al centro del campo de fuerza central, y theta es su ángulo polar con respecto a una semi-recta de referencia, que parte del origen de coordenadas y se extiende hacia "su derecha". Para tratar de medir la velocidad radial instantánea de la partícula usamos la variable *vr*, y para medir su velocidad angular utilizamos omega. La constante k es la que aparece en la ecuación (1), y se llama constante de resorte, o constante de Hooke. Las coordenadas polares se pueden transladar a coordenadas rectangulares y para eso se utilizan las variables *x* y *y*. Las cuales tendrán las siguientes relaciones con las coordenadas polares:

$$\begin{aligned} x &= r\cos(\theta), \\ y &= r\sin(\theta). \end{aligned} \tag{2}$$

La variable *l* representa a el momento angular de la partícula, y según la teoría toma una expresión constante dada por la siguiente relación

$$\begin{aligned} l &= \omega m r^2, \\ l &= m r^2 \dot{\theta} = m r^2 \frac{d\theta}{dt}. \end{aligned} \tag{3}$$

Y también un valor constante, durante todo el movimiento.

TABLA I. Variables de la simulación y sus valores iniciales

Variables	
Nombre	Valor inicial
t	0.0
h	0.01
m	1.0
l	1.0
theta	1.7
r	1.2
vr	1.4
x	r*Math.cos(theta)
y	r*Math.sin(theta)
k	1.0
omega	l/(m*r*r)

Para ver como editar esto en la ventana de variables de Easy Java Simulations podemos ver esto en la figura 2.

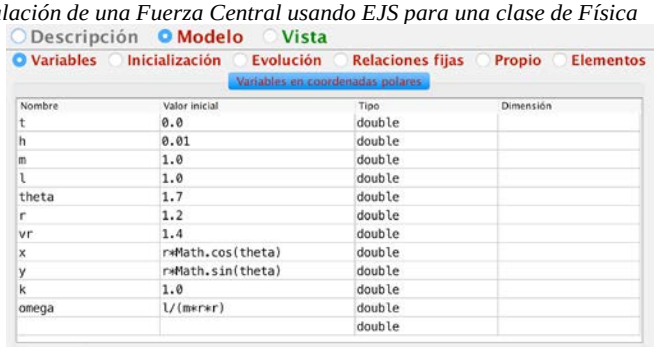

FIGURA 2. Lista de variables para la simulación de una partícula en un campo de fuerza central que sigue la ley de Hooke.

En la figura podemos apreciar que todas las variables son de tipo de doble precisión, que tiene un valor inicial y que pueden tomar dimensiones. Sin embargo, y para no complicar mucho las cosas hemos preferido no darle dimensiones a las variables y manejar sólo sus valores numéricos, esto nos simplificará el código, y por supuesto podemos asumir un sistema estándar de unidades como el SI (Sistema Internacional).

V. ECUACIONES DE MOVIMIENTO

Con el objeto de realizar la etapa de la evolución del movimiento, es muy importante entender las ecuaciones de movimiento que rigen el comportamiento del sistema. Una de las ecuaciones de evolución está dada por la aceleración radial que sería

$$f_1(t, v) = \frac{dv}{dt} = \frac{d\dot{r}}{dt} = \ddot{r} = -\frac{k}{mr} + \frac{l^2}{mr^3}. \tag{4}$$

Otra es la sencilla relación, auto-definitoria de velocidad angular

$$f_2(t, \theta) = \omega = \frac{d\theta}{dt} = \dot{\theta}. \tag{5}$$

Ambos definen nuevas funciones en términos de varias variables. La metodología consiste en despejar una derivada en una de las coordenadas y reconocerla como una nueva función de las variables involucradas.

VI. MÉTODO DEL PUNTO MEDIO

El método es de segundo grado en exactitud y es bastante bueno para la aproximación de la evolución del movimiento. El método numérico, consiste en dividir el tiempo en una secuencia de valores

$$t_0, t_1, t_2, ..., t_i, ..., t_n. \tag{6}$$

La diferencia entre un par de valores y el siguiente se mantiene constante

$$t_1 - t_0 = ... = t_i - t_{i-1} = ... = t_n - t_{n-1} = h. \tag{7}$$

A su vez, las diversas variables que dependan del tiempo también sufren una discretización de valores de la manera de secuencia, por ejemplo para la velocidad radial tendremos la serie

$$v_{r0}, v_{r1}, v_{r2}, \ldots, v_m \tag{8}$$

Dado este esquema discreto de valores es necesario hallar expresiones que aproximen numéricamente los valores discretos de funciones como las mostradas en (4) o (5).

El método más sencillo es el de Euler quien hace la aproximación siguiente para la primer derivada y que es una función del tiempo y de la coordenada relacionada a la derivada. Así:

$$\begin{aligned} &v_{r0}, \\ &v_{r1} = v_{r0} + h\, f_1(t_0, v_{r0}), \\ &v_{r2} = v_{r1} + h\, f_1(t_1, v_{r1}), \\ &\ldots \\ &v_{r,n+1} = v_{r,n} + h\, f_1(t_n, v_{r,n}). \end{aligned} \tag{9}$$

Este es un método de primer orden, y tiene como consecuencia no ser tan aproximado a los valores reales de la secuencia de valores *v*. Para mejorar su aproximación se toma el método de segundo orden del punto medio o método de Euler modificado. Consiste precisamente en tomar un valor de *en medio,* del intervalo entre t_n y t_{n+1} . se toma el punto

$$t_{n+1/2} = t_n + \frac{h}{2}. \tag{10}$$

y la aproximación de Euler para la función *f* a aproximar:

$$v_{n+1/2} = v_n + \frac{h}{2} f_1(t_n, v_n). \tag{11}$$

El método del punto medio consiste en tomar este valor aproximado para calcular el nuevo valor v_{n+1}. De esta forma

$$\begin{aligned} &v_0, \\ &\ldots \\ &v_{n+1} = v_n + h\, f_1(t_{n+1/2}, v_{n+1/2}). \end{aligned} \tag{12}$$

O bien, esto se reescribe como

$$\begin{aligned} &v_0, \\ &\ldots \\ &v_{n+1} = v_n + h\, f_1\left(t_n + \frac{h}{2}, v_n + \frac{h}{2} f_1(t_n, v_n)\right). \end{aligned} \tag{13}$$

Donde obviamente f_1, es la función que aparece en (4) para la partícula en un campo de fuerza central que sigue la ley de Hooke, y para la coordenada radial *r*.

Tendremos una fórmula similar para la otra función f_2, para la coordenada angular theta.

$$\begin{aligned} &\theta_0, \\ &\theta_1 = \theta_0 + h f_2\left(t + \frac{h}{2}, \frac{h}{2} f_2(t, \theta, \ldots, r, \ldots)\right) \\ &\ldots \\ &\theta_{n+1} = \theta_n + h f_2\left(t + \frac{h}{2}, \frac{h}{2} f_2(t_n, \theta_n, \ldots, r_n, \ldots)\right) \end{aligned} \tag{14}$$

Y de esta manera se obtiene una *evolución* para las variables velocidad radial $v=v_r=vr$, y también de la coordenada angular theta. Habrá otras identidades similares para otras variables.

VIII. EVOLUCIÓN

Ya que tenemos la forma que debe tener la aproximación a varias variables utilizando el método del punto medio, hay que aplicarla a la ventana de Evolución de EJS, utilizando el lenguaje de programación Java [4-11]. El código sería de la siguiente forma:

```
omega = l/(m*r*r);
theta = theta + h*omega;

double vr0 = vr;
vr = vr + h/2 * (-k/m*r + l*l/(m*r*r*r));
double r_N = r + h*vr;
double r_M = r + h/2*vr0;
vr = vr0 + h*(-k/m*r + l*l/(m*r_M*r_M*r_M));

r=r_N;
t =t + h;
```

Y además de este código que evoluciona a nuestras variables existen relaciones fijas entre las variables dadas por el cambio de coordenadas polares a coordenadas cartesianas dadas por las ecuaciones dadas en (2).

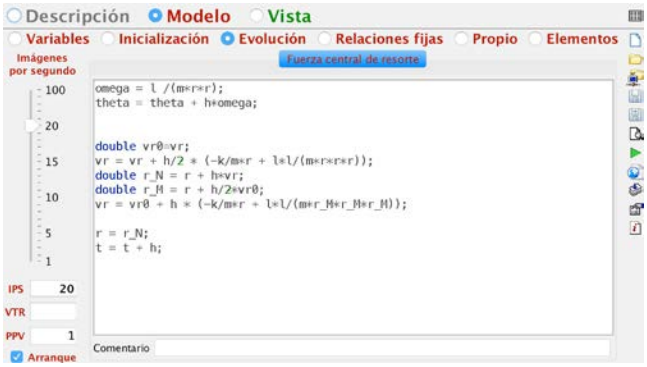

FIGURA 3. Código para la evolución, tomando en cuenta el método del punto medio dada por las relaciones (13) o (14).

En la figura 3, mostramos como se puede ingresar este código Java a la ventana Evolución de EJS. Sin variar otros parámetros como el número de imágenes por segundo, podemos dejar las cosas de esta forma.

Algunas aclaraciones serían pertinentes aquí, y es que la variable vr0, es la velocidad radial inicial al paso evolutivo, r_N se refiere al enésimo punto de *r*, y r_M sería el punto que está en medio. La expresión $t = t + h$, es para que el tiempo evolucione o pase a su siguiente valor. Si se nota algunas expresiones utilizan el tiempo avanzado a la mitad de su intervalo.

IX. RELACIONES FIJAS

De acuerdo a lo que hemos discutido, las relaciones fijas para esta simulación serían las que expresan el cambio de coordenadas del sistema polar al sistema cartesiano, y que en código Java, se escribe como

x = r * Math.cos(theta),
y = r * Math.sin(theta).

Este código se ingresa en el espacio de edición para la sub ventana de Relaciones fijas, como se hizo en el caso de la Evolución.

X. VISTA

La Vista es la parte de la simulación que es visible al usuario y tiene una representación del fenómeno Físico a parte de una o varias gráficas para sus variables. En este caso vamos a tener cuatro ventanas. Una de ellas es donde se encuentra la partícula sometida al campo de fuerzas centrales. Otra ventana os mostrará como evolucionan las coordenadas cartesianas de la partícula, Una tercer ventana nos ayudará a visualizar el movimiento desde sus coordenadas polares y una cuarta gráfica nos mostrará dos gráficas del espacio de fases para la partícula.

En la parte de la Vista se disponen de dos columnas o espacios verticales de edición uno a la izquierda y otro a la derecha de la ventana.

El espacio de la izquierda se llama *Árbol de elementos*, y es donde se tiene el control de todas las partes gráficas que componen la visualización de la simulación, y es la parte editable de la Vista.

El espacio vertical derecho lleva por nombre *Elementos para la vista*, y consta de todas las partes visuales o bloques de construcción de que dispone EJS para ir armando a manera de elementos gráficos, las partes esenciales de toda Vista. Es decir, la parte derecha, no es editable y sólo está ahí para que el usuario decida que recursos gráficos tendrá su simulación, estos recursos una vez seleccionados se disponen en una estructura de datos de árbol en la parte izquierda de la ventana.

De esta forma, seleccionamos algún elemento que hay en la parte derecha y la llevamos a la parte izquierda para ir forman un árbol de partes gráficas que van a componer la Vista de la simulación. Cuando la parte izquierda queda finalizada, entonces habremos acabado de diseñar la vista de la simulación. La parte izquierda será un Árbol que tendrá cada parte de la simulación. Por ejemplo, La parte izquierda ya consta con un elemento raíz de default que en EJS se llama *Vista de la simulación.* A este elemento le vamos agregando las ventanas que hemos propuesto anteriormente a manera de hijos de la raíz. Así que en nuestro diseño tendremos la raíz de la Vista con cuatro elementos que son las ventanas principales donde iremos editando las partes visuales de la simulación, y así las ventanas servirán de contenedores de las demás partes de la simulación. Las cuatro ventanas que proponemos tendrán los siguientes nombres

- Partícula
- Frame_X_Y
- Frame_R_THETA
- Espacio de fases

En la ventana de la simulación llamada Partícula estará la animación principal del cuerpo sometido al campo de fuerza central, dispondrá de botones de control como Play y Pause, o Reset, y nos dibujará la trayectoria seguida por el cuerpo de prueba, cuando está influenciado por el campo de fuerza central. Es importante que el estudiante revise esta parte y saque sus conclusiones.

En la ventana llamada Frame_X_Y hay un PanelConEjes que alberga a dos Rastros que representan la evolución de las coordenadas cartesianas *x* y *y* de la partícula en movimiento, con respecto al tiempo.

En la ventana llamada Frame_R_THETA, tendremos unos diagramas similares a los de la pasada ventana pero para las coordenadas polares *r* y theta, junto con sus respectivas velocidades instantáneas *vr* y omega. Así que aquí dispondremos de cuatro gráficas.

En la ventana llamada Espacio de fases, dispondremos de dos gráficas o Rastros, uno de ellos es la gráfica de la fase para la coordenada *r*, es decir es una gráfica cuyo componente horizontal es *r* y componente vertical la velocidad radial *vr*. La gráfica la escogemos en color azul. La segunda gráfica representa su diagrama de fase para la coordenada angular theta, y va en color rojo. Para esta gráfica el eje horizontal representa la coordenada angular theta y el eje vertical representa la velocidad angular omega.

La forma de llevar elementos de la columna derecha de la ventana Vista, a la raíz de la columna izquierda es un método genérico a todo elemento gráfico y lo podemos entender con un sólo ejemplo: Supongamos que queremos añadir la primer ventana Partícula a la raíz *Vista de la simulación* de la columna izquierda. Para ello primero elegimos una Ventana de la columna derecha de la ventana y luego ya seleccionada hacemos un click izquierdo de ratón a la raíz de la ventana o columna de la izquierda, trayendo así la ventana de la columna derecha a la columna izquierda, algo así como si arrastráramos el elemento de la parte derecha a la parte izquierda de la ventana Vista. Este procedimiento se repite una y otra vez con cada elemento del Árbol de la vista, hasta que este queda totalmente construido según nuestro diseño particular.

En la figura 4 mostramos como queda el árbol padre de la columna izquierda cuando seleccionamos cuatro ventanas de la columna Elementos para la vista del lado derecho, y los traemos a la raíz Vista de la simulación de la columna izquierda. Incidentalmente las ventanas se encuentran en el primer bloque de la columna derecha llamada Interfaz, para el segundo botón de menú superior llamado *Ventanas contenedores y paneles de dibujo*, que ha sido seleccionado el elemento se llama *Ventana: Una ventana de pantalla*, y tiene un recuadro morado alrededor en la figura indicando que lo hemos seleccionado previamente.

FIGURA 4. Cuatro ventanas son los elementos gráficos principales de la simulación y están como hijos de la raíz Vista de la simulación.

Para acabar de editar la Vista de la simulación se van ir añadiendo poco a poco varios hijos a estas cuatro ventanas para ir construyendo la simulación completa. Una vez que

hallamos terminado con este paso, se procede a editar las *Propiedades*, de cada elemento, permitiendo que la vista funcione y este ligada a las variables dinámicas descritas en la fase del Modelo.

En lo que queda del artículo enlistaremos los elementos faltantes y sus listas de propiedades, finalizado de esta manera con la Vista de la simulación, y terminando de editar toda la simulación en sí. Nos restará mostrar como luce la animación ya acabada y funcionando, para que los estudiantes aprovechen sus características y aprendan la Física de una manera amena y didáctica.

XI. EDICIÓN DE LA VENTANA DE LA ANIMACIÓN DE LA PARTÍCULA

Para la ventana principal de la animación, llamada Partícula tenemos dos hijos principales tenemos un *PanelDibujo* y un *Panel* cuyos nombres respectivos son

- Fza_Central
- buttonsPanel

En Fza:Central están los elementos de la animación más importantes que son la partícula o cuerpo y su trayectoria

- cuerpo
- trayectoria

El cuerpo es una *Forma: Una forma 2D (elipse, rectángulo, …)*, y sirve para dibujar un pequeño cuerpo o partícula de forma circular, que se estará moviendo durante la animación en una órbita acotada y cerrada, según la teoría de la Mecánica Clásica, esto por encontrarse bajo la influencia de un campo de fuerza central que actúa como resorte, o que sigue la ley de Hooke según la ecuación (1).

La trayectoria es un *Rastro: Un rastro 2D*. Y simplemente lo usaremos para ir dibujando la trayectoria que va trazando la partícula de prueba durante su movimiento.

Para el Panel buttonsPanel, tendremos cuatro hijos

- playPauseButton
- resetButton
- aFieldLabel
- aField

Los dos primeros son botones de control, mientras que *aFiledLabel* es un campo informativo con la nota de que el siguiente elemento denotará el tiempo transcurrido, así que este elemento es puramente informativo, pero importante. El último elemento *aField* es un *CampoNumérico* que contendrá la variable de tiempo *t*. Y que lo ponemos ahí para que el lector tome nota de que momento de la evolución del movimiento se está observando, cambia continuamente de acuerdo al valor que tenga el tiempo en ese momento de observación.

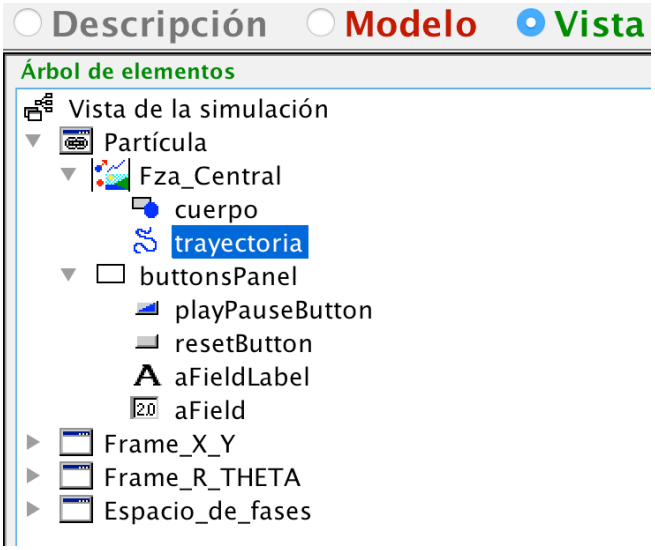

FIGURA 5. Ventana principal y todos sus hijos, traídos de la columna derecha *Elementos para la vista*. La animación principal consta de *cuerpo* y *trayectoria* como hijos principales.

La figura 5 muestra a la primer ventana con todos sus hijos, para entender esto que acabamos de discutir.

Ahora nos falta enumerar las características de cada elemento de la figura 5 para que nuestra descripción quede completa. Pero antes, nos conviene describir con un ejemplo como es que podemos editar las Propiedades de cada uno de los elementos gráficos que forma el Árbol de la Vista de una simulación, basta con que tengamos un ejemplo gráfico para tener la idea principal. Y luego podemos editar los demás elementos de una forma similar a la del primer ejemplo.

Por ejemplo, supongamos que queremos editar las propiedades de la ventana principal Partícula, entonces procedemos de la siguiente forma:

- Hacemos un click derecho de ratón sobre el elemento.
- Se abre un menú contextual.
- Elegimos la opción Propiedades del Menú contextual.
- Se abre una ventana de Propiedades particular al elemento que hayamos seleccionado en el primer paso.
- Nos disponemos a editar los varios campos que se encuentran en la ventana Propiedades del elemento que estemos trabajando.
- Una vez que acabemos de editar todas las propiedades deseadas, cerramos la ventana.

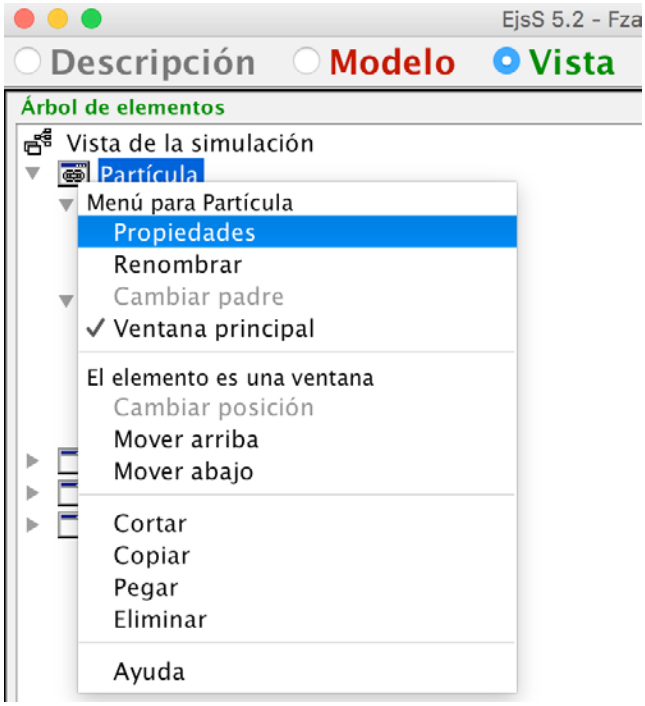

FIGURA 6. Menú contextual para un elemento gráfico, se selecciona la opción propiedades para editar la ventana del mismo nombre, del elemento respectivo.

Este proceso se repite una y otra vez para cada uno de nuestros elementos. Dependiendo de la cantidad de elementos gráficos que tengamos para una Vista en particular esto puede ser muy tedioso y repetitivo. Por lo tato hay que hacer un diseño, de acuerdo a las osas que deseemos ver de la simulación, entre más características mostremos, más largo y demandante será el proceso de añadir elementos a la raíz y editar sus propiedades respectivas. Así que en lo que sigue trataremos de ser breves y concisos para no desperdiciar espacio. Pero teniendo en cuenta una idea, que puede ser buena desde el punto de vista didáctico para el estudiante.

En la figura 6 se muestra el menú contextual para el elemento gráfico Partícula de nuestro ejemplo.

Se nos abre la ventana de Propiedades de Partícula (Ventana) y se pueden editar los campos que ahí se encuentran. En la figura 7, mostramos un ejemplo típico de llenado de una ventana propiedades, los campos editables se encuentran en tres grupos de edición que en este caso son

- Principal
- Posición y Tamaño
- Aspecto Gráfico

Y en la figura 7 se muestran los títulos en color azul, después vienen los campos en sí, y sus valores que debemos de llenar con cuidado, pues algunos campos son primordiales para que la Vista funcione correctamente.

La forma de llenarlo lo resumimos en la siguiente lista descriptiva:

Partícula:
Propiedades de Partícula (Ventana):
Principal:
Título: “Fza Central”
Distribución: border
Visible: true
Posición y Tamaño:
Posición: “789,23”
Tamaño: “292,287”

En esta lista sólo mostramos, las secciones principales y los campos que están llenos, omitimos aquellos campos vacíos para mostrar sencillez en la descripción y para no gastar espacio valioso. Hay que compara esta lista con la figura 7, y hacer la identificación de los campos que han sido editados, pues los demás elementos, se editarán de forma similar a este ejemplo.

FIGURA 7. Ventana *Propiedades de Partícula (Ventana)*, mostrando como se editan ciertos campos, para describir las propiedades de la ventana principal de la simulación.

Armados con este conocimiento preliminar podemos describir las demás partes que nos quedan de la ventana Partícula y luego los demás elementos de la Vista. En primer lugar , para el *PanelDibujo* Fza_Central estas serían sus propiedades editadas:

Fza_Central:
Propiedades de Fza_Central (PanelDibujo):
Escalas:
Autoscala X: false
Autoscala Y: false
Mínimo X: -2.4
Máximo X: 2.4
Mínimo Y: -2.4
Máximo Y: 2.4
Aspecto Gráfico:
Fondo: 200,220,208,255

Algunas propiedades como Fondo, que denota el color del fondo, pueden considerarse puramente ornamentalmente estéticas, y otras como Autoscala X, o Mínimo X, pueden ser consideradas propiedades modificables de acuerdo a las necesidades particulares que se tengan, o hasta de las características físicas de la pantalla de la computadora donde se corra la simulación. Sin embargo, este es un ejemplo, que sirve de modelo para editar las propiedades de otros elementos gráficos.

Para el Rastro llamado *trayectoria* estas serían las propiedades editadas:

trayectoria:
Propiedades de trayectoria (Rastro):
Entrada:
Entrada X: *x*
Entrada Y: *y*

Aspecto Gráfico:
Max Puntos: 600
Limpiar: false
No Repetir: true
Conectar: true
Color Línea: BLUE

Algunas de estas propiedades, definen de manera única el tipo de trayectoria que dibujará la partícula bajo el campo de fuerza central, estas son Entrada X, y Entrada Y, que ligan a las coordenadas cartesianas de la partícula *x* y *y*, con la animación principal de la simulación.

Asimismo colocamos las propiedades editadas de la Forma *cuerpo*, que representa la partícula bajo el campo de fuerzas centrales

cuerpo:
Propiedades de cuerpo (Forma):
Posición y Tamaño:
Pos X: *x*
Pos Y: *y*
Aspecto Gráfico:
Color Relleno: RED
Ancho Línea: 0.4

Las propiedades más importantes son Pos X, y Pos Y que ligan otra vez a las coordenadas cartesianas de posición de la partícula animada. Estas son las propiedades esenciales de la simulación. Las otras son opcionales, como el color de relleno de la partícula, y el grueso de su línea de borde.

Siguiendo la figura 5, debemos colocar las propiedades del Panel siguiente contenedor de los botones de control:

buttosPanel:
Propiedades de buttonsPanel (Panel):
Principal:
Distribución: FLOW:center,0,2
Bordes:
Tipo Borde: LOWERED_ETCHED

Continuamos con el botón que controla la corrida y la pausa de la animación:

playPauseButton
Propiedades de playPauseButton (BotonDosEstados):
Principal:
Variable: _isPaused
Propiedades Sí:
Imagen Si:
/org/opensourcephysics/resources/controls/imagens/play.gif
Acción Si: _play()
Propiedades No:
Imagen No:
/org/opensourcephysics/resources/controls/imagens/pause.g
if
Acción No: _pause()

Aquí las propiedades más importantes, son el tipo de variable de estado que se manipula que es _isPaused, y las dos acciones que se deben de ejecutar cuando se pulsa algún botón como la función _play() par poner en marcha la animación, o la acción _pause(), que sirve para pausar la animación. Otras propiedades más señalativas son la imágenes que se emplean para indicar el estado del botón como son play.gif y pause.gif, que se indican con todo a la ruta de acceso a la imagen

Luego siguen las propiedades del botón de regreso:

resetButton:
Propiedades de resetButton (Boton):
Principal:
Imagen:
/org/opensourcephysics/resources/controls/imagens/reset.gi f
Acción: _reset()

El aFieldLabel, es una etiqueta para indicar que en el siguiente campo se muestra el tiempo transcurrido de la animación

aFieldLabel:
Propiedades de aFieldLabel (Etiqueta):
Texto: " t="

Y el campo numérico donde se despliega el tiempo de animación transcurrido:

aField:
Propiedades de aField (CampoNumerico):
Principal:
Variable: *t*

Este es el último elemento hijo de la ventana de animación principal. Con lo cual hemos terminado de describir como se construye la animación principal.

En la siguiente sección vamos a describir como editamos una de las ventanas gráficas de la animación: aquella que nos muestra la evolución de las coordenadas cartesianas de la partícula con el tiempo.

XII. VENTANA DE LAS COORDENADAS CARTESIANAS DE LA PARTÍCULA

A la ventana que hemos dispuesto para mostrar la evolución de las coordenadas cartesianas de la partícula con el tiempo *t*, le hemos añadido los siguientes hijos que sigue una estructura de árbol:

frameXY:
  plottingXY:
    rastro_x
    rastro_y
  etiqueta_XY
    etiquetas_XY

Es decir, la ventana frameXY tiene a los hijos directos plottingXY que es un *PanelConEjes*, cuya función es disponer de los ejes de gráfica para trazar cada gráfica. Y también un Panel: Un panel contenedor básico, llamado etiqueta_XY, cuya función es contener unas etiquetas informativas o descriptivas de las gráficas. Las gráficas en sí vienen a estar proporcionadas por dos Rastros: rastro_x y rastro_y, que tienen la función de trazar las gráficas de evolución con el tiempo para las coordenadas cartesianas de la partícula, que son las representadas por las variables *x* y *y* de la animación. El elemento etiquetas_XY es solamente una etiqueta decorativa, que nos informará acerca de las dos gráficas trazadas en esta ventana.

En la figura 8, se muestra la estructura de árbol d elementos para la ventana frame_XY de la simulación.

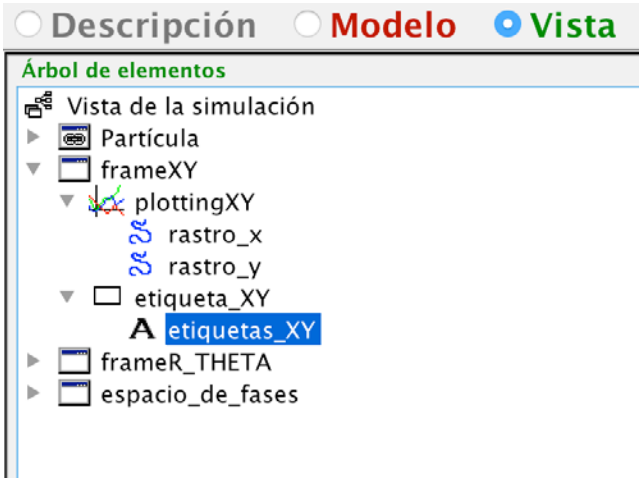

FIGURA 8. Elementos gráficos hijos de la ventana frameXY. Esta ventana contiene las gráficas rastro_X y rastro_Y, de la evolución de las coordenadas cartesianas para la partícula en movimiento dentro del campo de fuerzas central.

Entonces vamos a empezar enlistando las propiedades de la ventana contenedora frameXY

frameXY:
Propiedades de frameXY (Ventana):
Principal:
Título: "Frame-X-Y"
Distribución: border
Visible: true
Posición y Tamaño:
Posición: "249,57"
Tamaño: "300,300"

Luego describimos al PanelConEjes que dispone de la cuadricula donde van las gráficas y que se llama plottingXY. Estas son sus propiedades editadas que aparecen como un ejemplo posible:

plottingXY:
Propiedades de plottingXY (PanelConEjes):
Escalas:
Mínimo X: -0.3
Máximo X: 8.0
Mínimo Y: -2.0
Máximo Y: 2.0
Decoración:
Título: "posición X y Y"
Título X: "tiempo t"
Título Y: "coordenadas x y"
Configuración:
Fixed Gutters: false
Aspecto Gráfico:
Visible: true

Algunas propiedades como Mínimo X y Máximo X, dependen del rango de valores que adquieren las variables respectivas, por lo que es importante ir checando estos valores conforme hacemos algunas pruebas parciales de corrida de la simulación. Estos valores se editan hasta que tengan el rango adecuado.

Luego siguen las propiedades de la gráfica que muestra la evolución de la coordenada cartesiana *x* con el tiempo *t*, y que se llama rastro_x para EJS. Las propiedades pueden ser como siguen

rastro_x:
Propiedades de rastro_x (Rastro):
Entrada:
Entrada X. *t*
Entrada Y: *x*
Aspecto Gráfico:
No Repetir: true
Color Línea: BLUE

La gráfica queda entonces ligada con la variable coordenada *x* en Entrada X, y con el tiempo *t* en la propiedad Entrada Y. De esta forma, se construye una gráfica de *x* vs. *t*, que nos muestra la evolución de la variable coordenada de posición *x*, conforme pasa el tiempo de evolución de la animación medido por la variable independiente *t*. O dicho de otra forma, se logra hacer la gráfica de la función dependiente del tiempo

$$x = x(t). \tag{15}$$

En la figura 9, aparece como es editada esta gráfica.

FIGURA 9. Edición de una gráfica para la simulación. Corresponde a la gráfica que muestra la variación de la coordenada cartesiana *x* de la partícula en movimiento, con respecto al tiempo *t*. Representa a la gráfica de la función dada en (15).

La gráfica quedó trazada, en este ejemplo con una línea continua de color azul.

Luego hacemos lo mismo para la gráfica de la función coordenada

$$y = y(t). \tag{16}$$

Que corresponde al elemento gráfico rastro_y y que puede tener las siguientes propiedades:

rastro_y:
Propiedades de rastro_y (Rastro):
Entrada:
Entrada X: t
Entrada Y: y
Aspecto Gráfico:
No Repetir: true
Color Línea: RED

Luego, siguen las propiedades del Panel contenedor llamado etiqueta_XY, que contiene una etiqueta informativa acerca de las gráficas, como medio de aviso para el usuario.

etiqueta_XY:
Propiedades de etiqueta_XY (Panel)
Principal:
Distribución: border

La etiqueta propiamente dicha queda de la siguiente forma

Etiquetas_XY:
Propiedades de etiquetas_XY (Etiqueta):
Principal:
Texto: “ x<-azul y<-rojo t<-horizontal”

Como vemos el aviso suguiere que la gráfica que va de color azul representa la evolución de la coodenada *x* (ecuación (15)) con el tiempo *t*, la gráfica que va trazada en color rojo representa la evolución de la coordenada *y* (ecuación (16)) para el movimiento de la misma partícula con el tiempo *t*, y que el mismo tiempo *t* corre por el eje horizontal. Esta es una información visual valiosa para el usuario del programa que puede ser un estudiante o el mismo profesor de Física.

De esta manera, hemos concluído la descripción de cómo armar la segunda ventana de la simulación, la cual mustra información de evolución importante, acerca de las coordenadas cartesianas de la partícula en movimiento para el campo de fuerzas central. Esta es una de las ventnas que se pueden desplegar. Dependiendo de las necesidades de la clase se puede o no desplejar más información en otras ventanas diseñadas para tal fin.

En las secciones siguientes mostramos como construir y anotar las propiedades de otras dos ventanas, que despliegan información importante del fenómeno físico: Una ventana muestra la evolución de las coordenadas polares y de las velocidades polares instantáneas. La otra ventana muestra dos gráicas de fase para la partícula en el campo de fuerzas centrales.

Primero nos avocamos a la ventana que muestra la evolución de variables polares de la pártícula

XIII. VENTANA DE VARIABLES POLARES

Como ya hemos mostrado un ejemplo muy parecido en la sección pasada. Aquí vamos a mostrar como se puede construir y editar las propiedades de los elementos gráficos para una vemtana que muestre las gráicas de la evolución con el tiempo *t*, de varias variales asociadas a las coordenadas polares de la partícula en movimiento.

En la figura 10, mostramos el árbol de elementos gráficos para la ventana que muestra variables polares frameR_THETA.

A manera de consejo, debemos de seguir y respetar la estructura básica de árbol, ya que de esta forma los elementos gráficos se integran y colaboran entre sí de manera adecuada, sin presentar problemas de diseño. Así que si es importante siempre tener en cuenta la estructura de árbol de la Vista, y como algunos de los elementos, encajan como hijos de otros.

Entonces tenemos la siguiente estructura de árbol, de la figura 10:

frameR_THETA:
  plottingR_THETA
    rastro_r
    rastro_vr
    rastro_theta
    rastro_omega
  etiqueta_R_THETA
    etiquetas_r_y_vr
    etiquetas_theta_y_omega

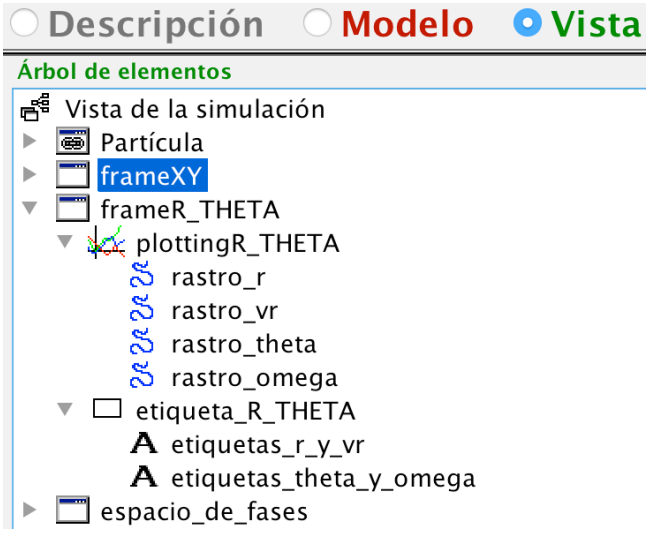

FIGURA 10. Estructura de árbol para la ventana frameR_THETA, que nos va a mostrar la evolución de varias variables asociadas a las coordenadasnpolares de la partícula en movimiento.

Siguiendo con nuestra práctica de enlistar las propiedades de cada elemento, empezamos por las propiedades de la ventana que contiene las gráficas que deseamos:

frameR_THETA:
Propiedades de frameR_THETA (PanelConEjes):
Principal:
Título: “frame-R-THETA”
Distribución: border
Visible: true
Posición y Tamaño:
Posición: “563,502”
Tamaño: “300,300”

Mostramos valores tentativos, que pueden cambiar de acuerdo a las necesidades y circunstancias específicas.

Luego, mostramos las propiedades del *PanelConEjes* que contiene a las gráficas y que traza el sistema de ejes gráfico:

plottingR_THETA:
Porpiedades de plottingR_THETA:
Escalas:
Mínimo X: -0.3
Máximo X: 8.0
Mínimo Y: -1.3
Máximo Y: 6.3
Decoración:
Título: “Posición R y THETA”
Título X: “tiempo t”
Título Y: “r vr theta omega”
Aspecto Gráfico:
Visible: true

Las cuatro gráficas son cuatro Rastros con las siguientes propiedades:

rastro_r:
Propiedades de ratro_r (Rastro):
Entrada:
Entrada X: *t*
Entrada Y: *r*
Aspecto Gráfico:

No Repetir: true
Color Línea: BLUE

Esta es la gráfica de la coordenada radial de la partícula que va en color azul. Para la velocidad radial tenemos las Propiedades del siguiente Rastro:

rastro_vr:
Propiedades de rastro_vr (Rastro):
Entrada:
Entrada X: *t*
Entrada Y: *vr*
Aspecto gráfico:
No Repetir: true
Color Línea: RED

La gráfica de la velocidad radial de la partícula irá en color rojo, como se ha sugerido aquí (aunque puede variar de acuerdo a los gustos personales).

Otra gráfica será la de la evolución de la coordenada angular theta, de la partícula. Esta será trazada por una gráfica que aquí se sugiere que vaya en color rosa. Las propiedades del Rastro que lo describe son las siguientes :

rastro_theta:
Propiedades de rastro_theta (Rastro):
Entrada:
Entrada X: *t*
Entrada Y: *theta*
Aspecto Gráfico:
No Repetir: true
Color Línea: PINK

Otro Rastro importante, tiene que ver con la gráfica para la velocidad angular de la partícula:

Rastro_omega:
Propiedades de rastro_omega (Rastro)
Entrada:
Entrada X: *t*
Entrada Y: *omega*
Aspecto Gráfico:
No Repetir: true
Color Línea: ORANGE

La gráfica se trazo en color naranja.

Continuamos con el Panel contenedor básico que tiene a las etiquetas informativas:

etiqueta_R_THETA:
Propiedades de etiqueta_R_THETA (Panel):
Principal:
Dsitribución: border

Y luego están las etiquetas informativas sobre la identificación de cada una de las gráficas que hemos puesto anteriormente.

etiquetas_r_y_vr:
Propiedades de etiquetas_r_y_vr (Etiqueta):
Principal:
Texto: "r<-azul vr<-rojo t<- horizontal"

etiquetas_theta_y_omega:
Propiedades de etiquetas_theta_y_omega (Etiqueta)
Principal:
Texto: theta<-rosa omega<-naranja"

Así estas etiquetas sugieren que la curva en azul representa la gráfica para la evolución de *r*, la gráfica azul corresponde a la velocidad radial vr, la gráfica rosa corresponde a la coordenada angular *theta*, la gráfica naranja es para la velocidad angular de la partícula *omega*, y finalmente el eje horizontal representa al tiempo *t*.

Las etiquetas se colocan una arriba de la otra, para una mejor distribución del espacio asignado para ellas.

XIV. VENTANA DE LOS ESPACIOS FASE

Finalmente, podemos describir como construir las gráficas de los espacio fase para la partícula. Hay una curva o gráfica para cada una de las coordenadas que se tengan. Como el problema es bidimensional, podemos escoger a las coordenadas polares como guías de graficación.

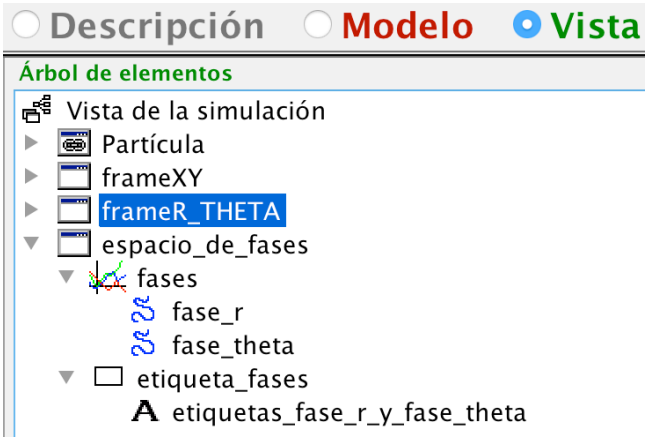

FIGURA 11. Ventana espacio_de_fases, mostrando a las dos gráficas que contiene y otros elementos decorativos.

Sin embargo, esta elección no es única y lo mismo valdría si hubieramos escogido a las coordenadas cartesianas. En lo que sigue suponemos que os interesa el espacio fase correspondientes a las curva de fase para cada una de las coordenadas polares de la partícula.

Una vez que escogemos que vamos a graficar, mostramos la ventana llamad espacio_de_fases y a sus hijos con forma de estructura de árbol, como ya nos es habitual:

espacio_de_fases:
  fases
    fase_r
    fase_theta
  etiqueta_fases
    etiquetas_fase_r_y_fase_theta

La figura 11, muestra esta misma estructura, pero en el editor de la Vista para EJS.

Ahora enlistamos las propiedades sugeridas para esta ventana contenedora:

espacio_de_fases:
Propiedades de espaio_de_fases (Ventana):
Principal:
Título: "espacio_de_fases"
Distribución: border
Visible: true
Posición y Tamaño:
Posición: "757,434"
Tamaño: "300,300"

Las propiedades del PanelConEjes llamado fases, y que despliega los ejes de trazo de las gráficas y que contiene a las gráficas son:

Fases:
Propiedades de fases (PanelConEjes):
Escalas:
Autoscala X: true
Auroscala Y: true
Mínimo X: 0
Mínimo Y: -1.6
Méximo Y: 3.8
Decoración:
Título: "fases"
Título X: " r theta"
Título Y: " vr omega"

Este panel contendrá a dos gráficas del espacio de fases. La primer gráfica es la correspondiente a la fase para la coordenada radial *r*. Sus propiedades serán las siguientes:

fase_r:
Propiedades de fase_r (Rastro):
Entrada:
Entrada X: *r*
Entrada Y: *vr*
Aspecto Gráfico:
No Repetir: true
Color Línea: BLUE

Así que las cantidades que modelan la fase son la coordenada radial *r* y su velocidad radial *vr*.

Luego. Sigue la segunda gráfica para la fase de la coordenada angular.

fase_theta:
Propiedades de fase_theta (Rastro):
Entrada:
Entrada X: theta
Entrada Y: omega
Aspecto Gráfico:
No Repetir: true
Color Línea: RED

Luego sigue la parte informativa, con su Panel contenedor

etiqueta_fases:
Propiedades de etiqueta_fases (Panel):
Principal:
Distribución: border

Ahora enlistamos las propiedades de la etiqueta que está adentro:

etiqueta_fase_r_y_fase_theta:
Propiedades de etiqueta_fase_r_y_fase_theta (Etiqueta):
Principal:
Texto: " fase r<-azul fase theta<- rojo"

Sugiriendo que la curva en color azul representa la gráfica de la fase radial, y la curva en color rojo representa la gráfica de la fase angular de la patícula. El estudiante puede observar y estidiar las características de cada una de estas fases. Se sugiere que las discutan con sus compañeros.

XV. CORRIENDO LA SIMULACIÓN

Es importate mencionar que Easy Java Simulations cuenta con varias funciones de control de edición de una simulación. Por ejemplo, podemos correr la simulación mientras la estamos editando en el programa. La ventana de edición cuenta con una barra vertical de herramientas en su lado derecho, como se puede observar de la figura 3. Si observamos con cuidado, hallamos el botón que tiene forma de triángulo acostado en color verde.

FIGURA 12. Botón play, para correr la simulación.

También tenemos un botón para guardar nuestro trabajo, en cualquier momento que lo requiramos.

FIGURA 13. Botón save, para guardar el trabajo

Además podemos guardar nuestro trabajo en formato autoejecutable *jar* de Java. Para esto contamos con el botón mostrado en la figura 14.

FIGURA 14. Botón de guardado en formato jar.

El formato *jar* se guarda en la carpeta de trabajo y en la carpeta export que esta adentro de la primera carpeta.

Para que el estudiante creativo pueda comparar su trabajo con el de nosotros en la figura 15, mostramos como quedó nuestro trabajo si corremos el código que hemos escrito aquí. La corrida fúé en un sistema operativo Mac OS X, pero la simulación puede ser hecha en Windows, Linux, FreeBSD, OpenBSD o cualquier otro sistema operativo compatible con Java.

XVI. METODOLOGÍA RECOMENDADA

La metodología didáctica recomendada para llevar a cabo la actividad de que se pueda editar en clase la simulación o al menos observar su ejecución es siguiendo los pasos de una Clase Demostrativa Interactiva [12- 6]. Según Sokoloff *et al*. la forma de hacer que el estudiante tenga activa su actitud y su comportamiento en clase, es mediante la utilización del ciclo PODS. El profesor propone integrar equipos de estudiantes de en promedio unos cuatro integrantes.

Esto se propone en cuatro etapas:

1. Etapa P de Predecir: El profesor pide al estudiante que escriba en una hoja en blanco sus predicciones.
2. Etapa O de Observar: El profesor corre la simulación y el estudiante compara el comportamiento con sus predicciones previas.
3. Etapa D de Discutir: El profesor les dice a los equipos que discutan entre sí sus experiencias y que llegen a alguna idea coherente.
4. Etapa S de Síntesis: Los equipos exponen su explicació del fenómeno físico estudiado, y si hay diferencias, entonces se discuten para llegar a un

acuerdo común, aquí el profesor guía la síntesis y puede proponer algunas ideas pero sin explicar el fenómeno en su totalidad. Más bien es como una guía para que los estudiantes puedan al fin construir su propio conocimiento.

A final cada estudiante modifica su hoja de predicciones previa de acuerdo a las nuevas ideas, y trata de explicar el fenómeno de mejor manera, y de acuerdo a lo discutido y observado. Esta metodología sería buena aplicarla una vez que se termine de editar la simulación. Pero durante su escritura, es importante que el estudiante trabaje, apoyado por la guía del profesor.

XVII. CONCLUSIONES

Esperamos que el trabajo aquí mostrado sirva de guía tanto para el profesor como para el estudiante que quieran construir una simulación de un fenómeo físico y así enriquecer la clase de Física proporcionando un recurso de viene de las Tecnologías de la Información y la Comunicación, y que es totalmente gratuito. Claro, tomando en cuenta que cada vez, las escuelas de bajos recursos pueden contar con unas salas de cómputo, o bien que cada estudiante puede contar con una computadora personal. Esto era antes muy difícil. Pero conforme avanza la tecnología y los avances que hace la ciencia en electrónica son más variados, parece ser que el mercado de las computaoras es cada vez más accesible, aún en países que están en vías de desarrollo tecnológico como lo es México, u otros países de Latinoamerica.

Si tomamos en cuenta esto, y el hecho de que el software presentado aquí es de fácil acceso y además es software libre. Creemos que tenemos buenos puntos a favor para recomendarlo.

El trabajo aquí mostrado entonces promueve que el estudiante tenga un comportamiento más activo hacia la clase de Física, mejorando con esto su propia formación académica, y provocando que participe mejor en su clase. Así sus conocimientos podrán ser ejemplares y más duraderos, que los obtenidos con una clase donde el estudiante sea pasivo.

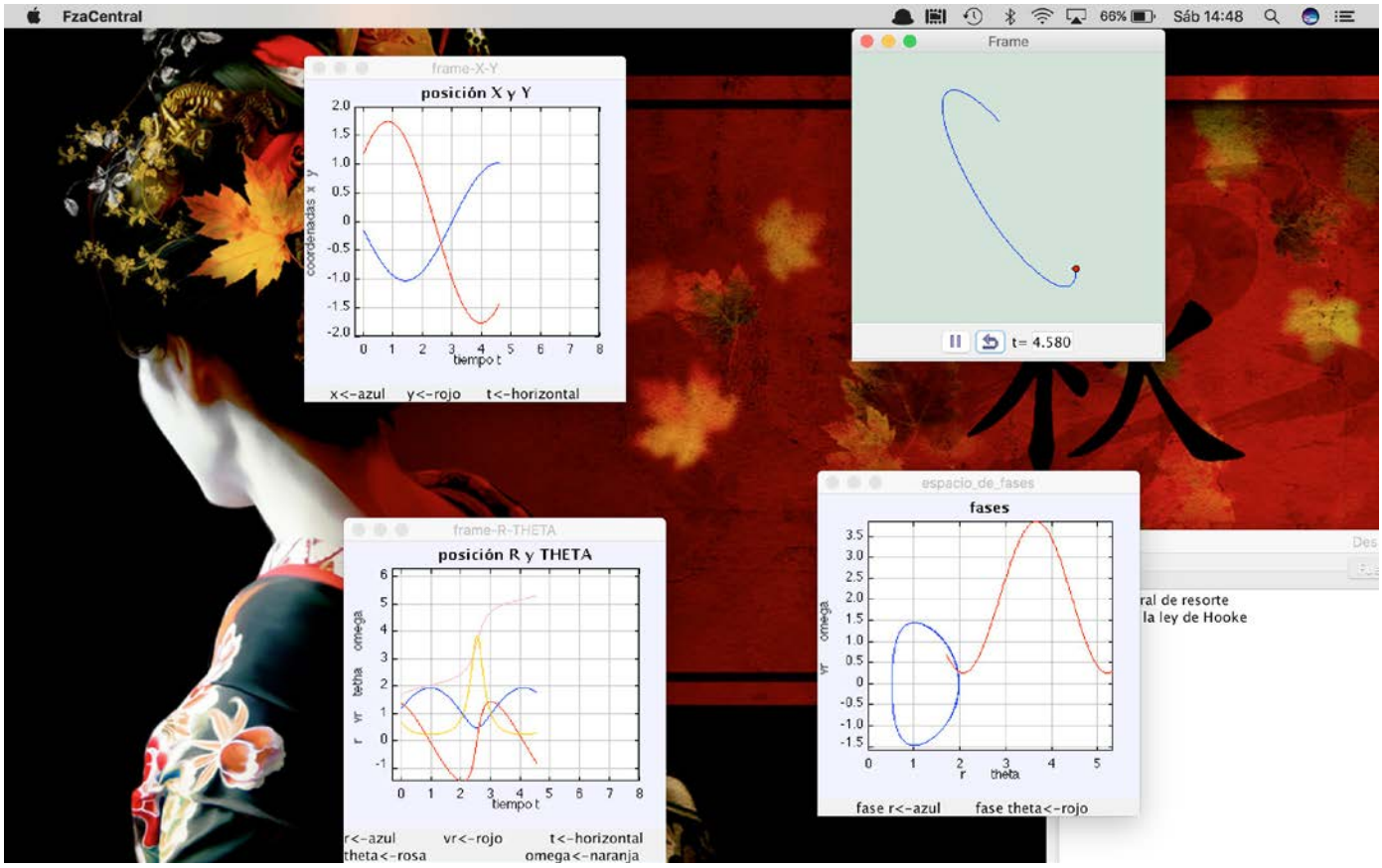

FIGURA 15. Corrida de la simulación para una partícula en un campo de fuerzas centrales que obedece la ley de Hooke.

AGRADECIMIENTOS

Se agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) de México, y al Sistema Nacional de Investigadores (SNI) de la misma dependencia. También el autor quiere agradecer el apoyo recibido por la COFAA del Instituto Politécnico Nacional. Gracias a estos apoyos el presente trabajo fue posible.

También se agradece al apoyo recibido del proyecto de la Secretaría de Investigación y Posgrado (SIP) del Instituto Politécnico Nacional de número de registro 20161695 y con título *Uso de prototipos experimentales para la enseñanza del movimiento rotacional en ingeniería – Diseño del test de evaluación*.

Esperamos desde luego, que el trabajo sea de utilidad a la comunidad de profesores y estudiantes de Física, para una mejor asimilación de los conceptos involucrados en el estudio de una fuerza central, actuando sobre un cuerpo de prueba, y cuya forma funcional es de una fuerza de resorte, que sigue la ley de Hooke.

REFERENCIAS

[1] Esquembre, F., *Creación de Simulaciones Interactivas en Java. Aplicación a la Enseñanza de la Física*. (Pearson Prentice Hall, Madrid, 2005).
[2] Esquembre, F., <http://fem.um.es/Ejs/>, Consultado el 02 de Mayo de 2016.
[3] Goldstein, H., *Classical Mechanics*, 2a Ed. (Addison-Wesley, NewYork, 1980).
[4] Bates, B., Sierra, K., *Head First Java*, 2a Ed. (O´Reilly & Associates Inc, California, USA, 2009).
[5] Niemeyer, P., Leuck, D., *Learning Java*, 4a Ed. (O'Reilly Media, 2013).
[6] Wadler, P., Naftalin, M., *Java Generics and Collections*, 1ª Ed., (O`Reilly &Associates Inc, California, USA, 2009).
[7] Rusty, E., *Java I/O*, 2ª Ed. (O'Reilly & Associates Inc, California, USA, 2010).
[8] Flanagan, D., *Java in a Nutshell, A Desktop Quick Reference*, 5ª Ed. (O`Reilly Media, USA, 2015).
[9] Flanagan, D., *Java Examples in a Nutshell, A tutorial Companion to Java in a Nutshell*, (O`Reilly & Associates Inc, Massachusetts, USA, 2004).
[10] Eubanks, B. D., *Wicked Cool Java*, (No Starch Press, San Francisco, 2005).
[11] Freeman, Eric, Freeman, Elisabeth, Sierra, K., Bates, B., *Head First Design Patterns*, (O`Reilly & Associates Inc, California, USA, 2004).
[12] Sokoloff, D. R., Thornton, R. K., *Interactive Lecture Demostrations*, *Active Learning in Introductory Physics*, *The Physics Suite*, 1ª Ed. (John Wiley & Sons, Inc., USA, 2006)
[13] Sokoloff, D. R., Thornton, R. K., Laws, P. W., *Realtime Physics Active Learning Laboratories, Module 1: Mechanics*, 3ª Ed. (John Wiley & Sons, Inc., USA, 2011).
[14] Sokoloff, D. R., Sokoloff, D., *Realtime Physics Active Learning Laboratories, Module 2: Heat and Thermodynamics*, 2ª Ed., (John Wiley & Sons, Inc., USA, 2011).
[15] Sokoloff, D. R., Laws, P. W., Thornton, R. K., *Realtime Physics Active Learning Laboratories, Module 3: Electric Circuits,* (John Wiley & Sons, Inc., USA, 2012).
[16] Sokoloff, D. R., Laws, P. W., Thornton, R. K., *Realtime Physics Active Learning Laboratories, Module 4: Light and Optics,* (John Wiley & Sons, Inc., USA, 2012).