



Propuesta de Aprendizaje Activo apoyado con una simulación del péndulo simple

Rubén Sánchez Sánchez, César Mora

Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada del Instituto Politécnico Nacional, Unidad Legaria, Legaria #694, Col. Irrigación, Del. Miguel Hidalgo, C.P. 11,500, México D. F.

ARTICLE INFO

Recibido: 12 Enero 2016
Aceptado: 05 Marzo 2016

Palabras clave:
Aprendizaje Activo,
Péndulo Simple,
Metodologías didácticas.

E-mail: rsanchezs@ipn.mx

ISSN 2007-9842

© 2016 Institute of Science Education.
All rights reserved

ABSTRACT

En este breve trabajo se expone la manera en como el aprendizaje activo combinado con el recurso del empleo de simulaciones pueden apoyar en la realización de una clase de Física, en particular se revisa la forma en como se puede obtener la simulación con uso de herramienta computacional gratuita y como se puede aproximar la evolución del fenómeno del movimiento del péndulo simple utilizando el método numérico de segundo orden conocido como método de Euler modificado. Aquí se propone que la exposición siga la metodología de una clase demostrativa interactiva para apoyar a los estudiantes mediante su propia participación en el desarrollo de la simulación y su ejecución. Esto permite una mayor comprensión de los métodos y del mismo fenómeno, y le permite ligarlo con las ecuaciones matemáticas que lo describen. Así el aprendizaje del alumno será completo y satisfactorio.

In this brief paper we discuss the way in which active learning combined with the use of simulations can support the realization of a physics class, in particular the way in which the simulation can be obtained using a computational tool Free and how the evolution of the phenomenon of simple pendulum movement can be approximated using the numerical method of the second order known as the modified Euler method. Here it is proposed that the exposition follows the methodology of an interactive demonstrative class to support the students through their own participation in the development of the simulation and its execution. This allows for a better understanding of methods and the phenomenon, and allows it to be linked to the mathematical equations that describe it. Thus the student's learning will be complete and satisfactory.

I. INTRODUCCIÓN

En tiempos recientes, la influencia que ha tenido el uso de TIC en educación se ha destacado por su gran adaptabilidad y practicidad la ayuda y medio de asistencia para facilitar el aprendizaje de la Física. Esta tecnología bien llevada con el uso de los principios del Aprendizaje Activo, pueden representar un medio eficaz para que los estudiantes de una clase de Física aprendan nuevos conceptos y se les facilite el trabajo que se lleva a lo largo de un curso de Física. En este artículo, el autor trata de mostrar como un ejemplo bien desarrollado de simulación de un fenómeno físico combinado con el ciclo PODS de Sokoloff *et al.* (2006; 2011a; 2011b; 2012a; 2012b) puede trazarse con el objeto de mejorar el aprendizaje de los conceptos del péndulo simple en una clase a nivel bachillerato o superior. También se puede aprovechar para mostrar las técnicas de aproximación que existen para los métodos numéricos empleados. De esta forma, el código aquí mostrado nos instruye no sólo a realizar actividades propias que involucren la creatividad para formar material de apoyo para los estudiantes, sino que también ellos pueden analizar como es que funciona un método matemático en la aproximación de la evolución del movimiento de un fenómeno físico, en este caso, en la evolución del movimiento del péndulo simple. La técnica numérica que se puede explorar es la conocida como del punto medio o método de Euler modificado (Esquembre, 2005).

En los siguientes párrafos se describe como podemos construir la simulación en clase. Esto, llevado en exposición como una clase ilustrativa colaborativa, apoyará al estudiante a no sólo comprender el fenómeno físico sino que también reforzará su comprensión en los métodos numéricos. La herramienta de software que utilizamos en este ejemplo y propuesta es la herramienta de software construida por Francisco Esquembre, para preparar simulaciones de Física, se llama Easy Java Simulations o EJS (Esquembre, 2005; 2016a) por sus siglas en inglés. Existe una página especializada de Easy Java Simulations donde el software se puede adquirir fácilmente, bajándolo de internet de manera totalmente gratuita. Es por eso que este tipo de software son muy recomendables en el apoyo a la labor docente, en particular para reforzar clases de Física (Esquembre, 2016b).

II. BREVE DESCRIPCIÓN DEL CICLO DE APRENDIZAJE ESTUDIANTIL

El ciclo PODS o ciclo Predecir, Observar, Discutir, y Sintetizar fue propuesto por Sokoloff, *et al.* (2006; 2011a; 2011b; 2012a; 2012b) para la implementación de su metodología didáctica conocida en la literatura como Aprendizaje Activo de la Física en conjunción con un escenario de Física real de laboratorio. Aquí la utilizamos, como parte esencial de nuestra exposición y para lograr que el estudiante tome parte activa de la clase. El profesor primero debe sugerir la integración de grupos de trabajo de no más de cuatro estudiantes por equipo, la finalidad de este prerrequisito es lograr que todo estudiante participe en el desarrollo de las actividades, y no crear miembros de equipo que sólo se dediquen a observar como los demás miembros del equipo participan en la clase. También se forman para coadyuvar a cada estudiante a formarse un criterio propio de lo observado durante el desarrollo de las actividades. La bibliografía de Sokoloff *et al.* nos muestran ocho pasos seguidos por los estudiantes y el profesor. Aquí retomamos el ciclo principal y las ideas de este ciclo, de tal forma que queden acoplados a nuestros objetivos particulares de presentar la clase mediante una simulación elaborada, condicionada y corrida por los mismos estudiantes. El objetivo: alcanzar a diseñar la simulación del movimiento del péndulo simple. Este tema es bastante común verlo en cursos de Física de Bachillerato y aún para alumnos que ya están en una carrera técnica o científica. Esto es, la metodología puede ser aplicada para estudiantes tanto de nivel vocacional (que ya hayan llevado algunas nociones de cálculo, aunque este requisito no es tan demandante como parece al principio), como de la carrera para Física o para Ingeniería. De hecho, aún para nivel Posgrado puede aplicar, para estudiante que sigan la carrera de Física, y estén viendo ejemplos que involucren el formalismo Lagrangiano o Hamiltoniano.

III. EL CICLO PODS ACOPLADO PARA LA SIMULACIÓN DEL PÉNDULO SIMPLE

El ciclo PODS consta esencialmente de cuatro etapas de desarrollo, que se pueden repetir (de ahí el nombre de ciclo), tantas veces como sea el caso, para que un estudiante alcance un nivel de comprensión adecuado del fenómeno físico estudiado, en este caso toca comprender la dinámica del movimiento del péndulo simple.

Los cuatro pasos del ciclo PODS son los siguientes.

1. El profesor le da a cada estudiante de cada equipo, de manera individual una hoja de papel, donde apuntará sus predicciones que tenga acerca del fenómeno. Claro que para esto, el profesor conduce la clase de tal forma que despierte la inquietud del estudiante, para que trate de explicar el fenómeno con sus propias palabras. Una breve descripción general del fenómeno del movimiento del péndulo simple, debe de bastar, para que el estudiante trate de sacar su propia predicción. A esta etapa se le conoce como la etapa de la “Predicción”.
2. En la siguiente etapa, el profesor ayuda con cierto material (que por cierto, en este trabajo describo más adelante), como es que se puede preparar una simulación para el caso del movimiento del péndulo simple. Además de que el estudiante puede escribir el código y comprender el método numérico que describe la situación física, es el estudiante quien debe de poner a prueba su trabajo y “correr” su simulación, ya preparada por él. De hecho aquí sería la etapa de la “Observación” del fenómeno físico, pero sustituida con el trabajo previo del código de simulación. El estudiante debe de trabajar en su propio equipo durante el desarrollo de esta etapa del ciclo PODS.
3. La siguiente etapa se conoce como “Discusión”. Aquí el estudiante, puede entrar en razonamiento profundo y crítico acerca del fenómeno que acaba de observar. Es en esta etapa, que el estudiante tiene la oportunidad de “construir” su

propio conocimiento, comparando su previa predicción, con el comportamiento del fenómeno, ya observado en la etapa previa. Para la construcción del conocimiento, de acuerdo a la teoría del constructivismo cimentada en la teoría del desarrollo cognitivo de Piaget (1969; 2003), el estudiante interactúa con un pequeño medio social de acuerdo a la teoría del constructivismo ya cimentada en el pensamiento de Vygotsky (Daniels, 2001; 2008; 2012). De esta manera, los estudiantes pueden discutir lo necesario, para llegar a una especie de pre-conclusión del fenómeno estudiado, en este caso se trata del movimiento del péndulo simple. El profesor fija un tiempo para esta etapa de las actividades y les pide a cada equipo que se preparen para que uno de sus integrantes, explique el fenómeno de movimiento del péndulo simple, a su manera.

4. No menos importante es la cuarta etapa del desarrollo de actividades por parte de los estudiantes, pero coordinadas con sugerencias no definitivas, pero si leves que tenga el profesor. Los equipos exponen sus ideas cada uno, mediante un representante que previamente eligieron en la etapa anterior. Si hay discrepancias en los equipos el profesor sugiere algunas ideas y coordina la actividad, con el objeto de “Guiar” y no “resolver” el razonamiento de los estudiantes, tratando de “comparar” los distintos razonamientos. Esto da pie a que el conocimiento sea construido por cada estudiante de la manera “correcta” o de la forma mas “atinada”.

Todas las etapas previas están pensadas para que el estudiante, participe activamente en su propio aprendizaje. Si por alguna razón de fuerza mayor, el método fracasa. Entonces, se vuelven a repetir, esta vez intercambiando a los miembros de los equipos de alguna forma diferente a la anterior, y se repite el todo el proceso, hasta que los estudiantes puedan llegar a “construir” su propio conocimiento. Como el proceso está pensado para repetirse, el proceso en si mismo constituye un “ciclo” de aprendizaje estudiantil, como lo pretendemos acoplar para el caso del aprendizaje del movimiento del péndulo simple.

Ya armados con estas ideas, procedemos en los siguientes párrafos a explicar un poco de código en Easy Java Simulations con el objeto de apoyar al profesor de Física, que vaya a utilizar esta propuesta didáctica.

IV. LA HERRAMIENTA DE SOFTWARE

Como ya lo habíamos mencionado antes, una de las herramientas mas recomendables para realizar el trabajo de escribir el código de la simulación es por ejemplo, Easy Java Simulations (EJS), pues es software que puede bajarse de manera libre sin generar costos. En el momento de escribir este trabajo la versión del software es la 5.2, y puede bajarse de su sitio web.

Un prerequisite a la utilización de EJS es la instalación del lenguaje de programación Java, en la computadora donde se trabaje la simulación, o más específicamente, se necesita la instalación de una máquina virtual de Java. Un JRE o Java Runtime Environment (por sus siglas en inglés), es suficiente para cumplir este objetivo previo. El paquete JRE (Oracle, 2016) puede bajarse de la página siguiente de Oracle:

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

Una vez que se baja el software, la instalación de Java es muy sencilla, y existen guías apropiadas de instalación, en la misma página de Oracle, de acuerdo al sistema operativo del computador donde se desea realizar la simulación del péndulo simple.

Cuando bajamos el software propia de EJS, lo que encontramos en nuestro directorio de “downloads” particular a nuestro sistema operativo, es simplemente una carpeta, cuyo nombre irá variando de acuerdo a la versión de EJS que hayamos bajado. Puede llamarse algo como EjsS_nnn. Donde la palabra “nnn” es precisamente el número de versión del software que nos ha tocado bajar de internet.

V. PREPARACIÓN PRELIMINAR

Lo que hay que hacer es buscar un programa llamado algo así como EjsConsole.jar. Los archivos tipo “jar” de Java pueden ser corridos como programas ejecutables, una vez que se ha instalado un JRE. Lo primero que nos va a pedir la consola de EJS cuando se abre por primera vez es que le indiquemos cual va a ser nuestro directorio de trabajo. Se puede aceptar el directorio de trabajo por defecto, como una buena opción. Sin embargo, dependiendo de los planos del autor de la simulación este directorio puede ubicarse en un lugar diferente. En mi caso, ya había hecho un directorio

aparte que nombre “workspace” en mi directorio raíz de usuario de mi Macintosh, y simplemente le dije a la consola que iba a utilizar este directorio de trabajo, en lugar del que viene por default. Mis razones personales, fueron que estimaba que como van a salir nuevas versiones de EJS, en un futuro, simplemente puedo “borrar” todo el directorio viejo donde estaba EJS, sin borrar mi trabajo, y reemplazar este directorio con uno nuevo que baje de Internet, en dado caso que el software se actualice. De esta forma, puedo “guardar” mi trabajo aparte, y borrar el software viejo, sin perder nada. Pero esto es al gusto del que vaya a escribir la simulación del péndulo simple.

VI. PREPARACIÓN DE LA SIMULACIÓN EN EJS

Para escribir una simulación en EJS, se parte de un esquema de tres partes:

1. En la primera parte, se escribe la “Descripción” del fenómeno que se vaya a estudiar. En nuestro caso particular podemos anotar la teoría del péndulo simple, incluir un diagrama descriptivo, y también mencionar a la ecuación diferencial que gobierna la evolución de movimiento.
2. La segunda parte, es el “Modelo” en sí del movimiento. Aquí el trabajo consiste identificar las variables que vamos a usar para el fenómeno y en escribir tanto la ecuación diferencial, como las restricciones que rigen el movimiento de nuestro péndulo simple. Así también como, dar los valores iniciales de nuestras variables.
3. La segunda parte, consiste en el diseño de la “Vista” de la simulación. En esta parte vamos añadiendo los elementos gráficos que van a ser mostrados cuando corremos nuestra simulación. Por ejemplo, para un péndulo simple. Necesitamos una cuerda de longitud fija, un techo en donde poner la cuerda y una masa fija al otro extremo de la cuerda. Todos estos objetos son los “elementos gráficos” de nuestra simulación. También necesitamos una ventana donde se van a reunir los elementos pasados, y posiblemente un contador de tiempo.

Para una revisión más completa de estas partes de la simulación, se puede consultar el libro de Francisco Esquembre (2005). Aquí describimos brevemente estas partes para comodidad en el presente escrito. Lo que pasa es que cada vez que arrancamos la consola de EJS para escribir una simulación, se nos muestran dos ventanas, una de ellas es la “consola”, y la otra lleva el nombre del propio programa EJS. Y es en esta ventana donde realizamos la mayor parte del trabajo, llamémosla así simplemente: “ventana de trabajo”.

Bueno, pues en la ventana de trabajo (mostrada en la figura 1), notamos que en la parte superior hay tres opciones exclusivas denominadas “Descripción”, “Modelo” y “Vista”. Entonces cada uno de estos “botones exclusivos”, representa una “parte” del trabajo de escritura de la simulación. Como ya habíamos antes discutido, en un párrafo anterior de este apartado. Podemos elegir cualquiera de estos tres botones para trabajar en una de las tres “partes” esenciales de nuestra simulación. Lo que sigue es una breve descripción de como se puede manejar cada una de estas “partes” de la simulación para el péndulo simple.

VII. ESCRITURA DE LA “DESCRIPCIÓN” DEL PÉNDULO SIMPLE

Esta parte de la simulación es muy sencilla, y consiste en reunir el material de la teoría del péndulo simple en una página web, que sera mostrada cada vez que se abra el archivo de la simulación. Básicamente sirve como fuente de consulta para el estudiante. Ya hemos mencionado como escribir esta parte en la sección pasada.

No hay que tener mucha preocupación por los detalles técnicos aquí, ya que básicamente estamos escribiendo en un editor para páginas web, y la mayoría de nosotros ya hemos estado familiarizándonos con dichos editors. La figura 1 muestra un cuadro principal donde poner el material teórico del péndulo simple, y en la parte superior hay un sencillo menú para manejar el tipo de letra y su tamaño, entre otras opciones (como las de ligar una palabra a un sitio web, por ejemplo).

La figura 1, solo es un ejemplo breve de edición de material que nos sirve de información preliminar teórica para comprender mejor, como funciona el movimiento del péndulo simple. Donde nosotros ponemos, la cantidad de información que queramos incluir en esta parte del archivo de la simulación.

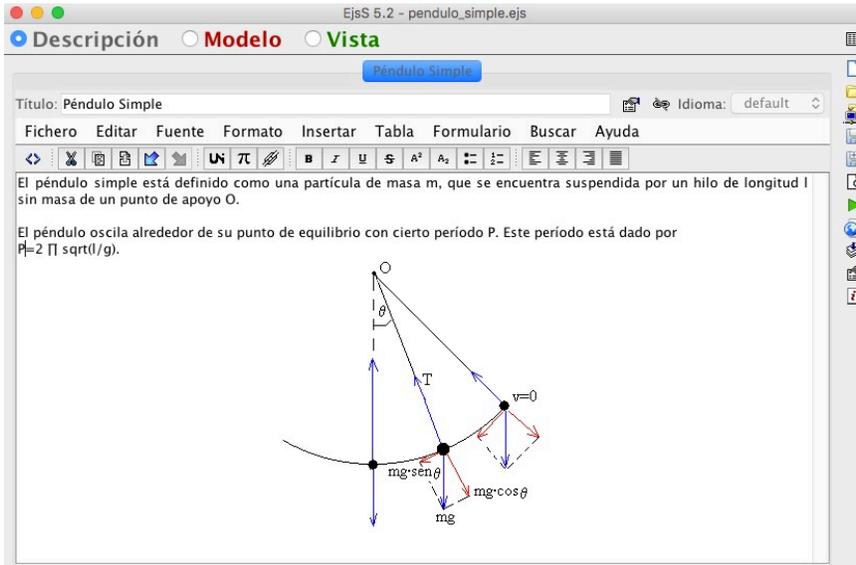


FIGURA 1. Desarrollo de la descripción, como se ve se edita con la información de la teoría del péndulo simple.

La forma general de utilización de Easy Java Simulations puede ser hallada y consultada en el libro de Francisco Esquembre (2005). Aquí se dan unas indicaciones más específicas para desarrollar la evolución del movimiento del péndulo simple, a partir de su ecuación diferencial (Goldstein, 1980).

$$ml \frac{d^2\theta}{dt^2} = -mg \sin(\theta), \quad (1)$$

donde hemos tenido en cuenta que la aceleración tangencial de la masa que cuelga esta dada por

$$a_t = l \frac{d^2\theta}{dt^2}. \quad (2)$$

VIII. EL MÉTODO MODIFICADO DE EULER

Para analizar la ecuación de movimiento del péndulo simple y tratarla numéricamente es necesario reescribir la ecuación (1), como el sistema de dos ecuaciones diferenciales de primer orden siguientes:

$$\begin{aligned} \frac{d\theta}{dt} &= \omega, \\ \frac{d\omega}{dt} &= -\frac{g}{l} \sin(\theta). \end{aligned} \quad (3)$$

En el método modificado de Euler se toma en cuenta una función del péndulo simple que es el segundo miembro de la segunda ecuación de (3) con una función de la variable dependiente $u := \omega = \omega(t)$, y la independiente t . Entonces, se tiene

$$f_2(\theta, t) = -\frac{g}{l} \sin(\theta). \quad (4)$$

El método del punto medio se puede expresar como

$$u_0 = \omega_0, \\ u_{n+1} = u_n + h f_2 \left(t_n + \frac{h}{2}, u_n + \frac{h}{2} f_2(t_n, u_n) \right). \quad (5)$$

Donde h representa “el paso” dt , entre un par de valores de las variables dependiente e independiente u_n, t_n , y el siguiente par de valores u_{n+1}, t_{n+1} .

Y para el segundo miembro de la primer ecuación en (3), tenemos las variables $v := \dot{\theta} = \dot{\theta}(t)$, y t . Y la siguiente identificación de función

$$f_1(\theta, t) = \omega. \quad (6)$$

Entonces el método modificado de Euler para la primer ecuación de (3) se expresa como

$$v_0 = \dot{\theta}_0, \\ v_{n+1} = v_n + h f_1 \left(t_n + \frac{h}{2}, v_n + \frac{h}{2} f_1(t_n, v_n) \right). \quad (7)$$

En estas ecuaciones debemos de entender que el método numérico nos da un par de variables dependiente e independiente u_n, t_n , del paso n a partir de los valores del paso pasado u_{n-1}, t_{n-1} . Desde luego que suponemos que contamos con los valores iniciales u_0, t_0 . Así que con las fórmulas obtenemos el par de valores u_1, t_1 , a partir del para anterior u_0, t_0 . También a su vez, obtenemos un para valores posterior u_2, t_2 , a partir ya de estos valores u_1, t_1 . Luego el proceso se repite para u_3, t_3 , y así sucesivamente hasta el para de valores u_n, t_n .

Una vez que hemos hecho estas identificaciones, en la siguiente sección mostramos la manera en como podemos escribir este método numérico, como código en Java dentro de EJS.

IX. CÓDIGO PARA LAS VARIABLES

En la ventana principal de EJS escojemos el radio botón marcado como “Modelo”, y después un sub radio botón que aparece bajo este llamado “Variables”. Nos aparecerá una ventana parecida a al mostrada en la figura 2. El objetivo del espacio de abajo, es introducir todas las variables, que en nuestro caso intervienen en el movimiento del péndulo simple. Además de especificar un valor inicial de las mismas. El arreglo de el espacio principal de la ventana es parecida a excel, y viene dotada con tres tipos de columna, la primera es “Nombre”, donde especificamos el nombre de la variable. La segunda es “Valor inicial”, donde colocamos un valor inicial de la variable, y que va afectar a la evolución de la simulación, pues controla a las condiciones iniciales de la ecuación diferencial. La tercer columna es nombrada “Tipo” y se refiere al tipo numérico de la variable, de acuerdo a los tipos de Java este, puede ser “double”, “int”, “float”, “short” o “long”, dependiendo de las necesidades que se tengan. Aunque generalmente el tipo más usado para variables de una simulación física es el double, es decir variables de la computadora que en el lenguaje de programación son de precisión doble y que cuentan con punto decimal. El número representado por Java ocupa 8 bytes de memoria. Los valores van de un mínimo de $1.8e-308$ hasta $1.8e308$, en valor absoluto. Otro tipo de valor numérico que puede usarse, al menos para los subíndices es el “int” que representa a un entero con signo en Java y ocupa 4 bytes de memoria los valores numéricos van desde -2147483648 hasta 2147483647 . Para mayor información sobre el sistema de números para Java puede consultarse la página web (Google, Tutorial Java, 2016)

<http://puntoconoesunlenguaje.blogspot.mx/2012/04/tipos-de-datos-java.html>

donde se hallan detalles de la implementación del sistema de números a través de los “tipos numéricos” de Java.

En la figura 2, mostramos un ejemplo de como luce una disposición de variables tipo excel para Easy Java simulations, en el caso del péndulo simple. Aquí, por ejemplo, tomamos un valor inicial del tiempo t igual a “cero” (0 seg), y se toma un incremento en los tiempos (que es la variable independiente) de $h = dt = 0.01$ (seg). Y Aunque las dimensiones son de segundos, como podemos observar se puede dejar la columna “Dimensión”, vacía sin grandes repercusiones. Basta con que nosotros recordemos que la dimensión correcta es de segundos. También colocamos una masa inicial de 1 (Kg), una longitud del hilo de 1 (m), una aceleración de la gravedad de 9.8 (m/seg²).

Un ángulo inicial del hilo con respecto a la vertical de Math.PI/12. Donde Math.PI, es el número matemático “ $\pi = 3.14159265\dots$ ”, en notación del lenguaje Java. Una velocidad angular “omega” de 0 (m/seg). Y realizamos un mapeo de coordenadas polares l , y ángulo “theta”, a las coordenadas cartesianas “ x ” y “ y ”. Estas últimas relaciones son “Fijas” y se mantiene así durante toda la evolución de movimiento del péndulo simple. Aquí se colocan en la subsección de “Variables”, aunque podrían colocarse en la de “Relaciones fijas”, sin ningún problema.

Después de la subsección de “Variables”, sigue la de “Inicialización”, sin embargo, como en este caso solo ocupamos la inicialización sencilla que hemos hecho de nuestras variables, y no ocupamos una ecuación de inicialización, más elaborada, podemos saltarnos esta subsección y procedemos con la subsección más importante de esta sección, que es la de “Evolución”, donde se colocan, en sí las ecuaciones diferenciales que rigen el movimiento del péndulo simple.

Así, que en la siguiente sección, nos dedicaremos a describir como se puede, escribir la ecuación diferencial (1) del movimiento del péndulo simple, de tal manera que podamos, escribirla en términos de aproximación numérica.

Desde luego, primero hay que sustituir el problema de la ecuación inicial de segundo orden, por el sistema equivalente de ecuaciones diferenciales de primer orden, escritas en (3). Este sistema, es más sencillo para anotarlo en forma numérica, que el problema original escrito en (1).

Un paso notacional, pero importante, es identificar a las funciones numéricas f_2 de (4) y f_1 de (6), que se necesitan para escribir el código final, en lenguaje Java.

X. CÓDIGO PARA LA EVOLUCIÓN DEL PÉNDULO SIMPLE

Como este trabajo es para ilustrar como se puede manejar, una aproximación numérica, nos avocaremos en describir al sistema de ecuaciones diferenciales de primer orden (3), en forma numérica con las ecuaciones (5) y (7).

Teniendo como funciones numéricas a la definidas en (4) por un lado y (6) por el otro.

Tenemos que escribir el código Java (Bates, Sierra, 2009; Nimeyer, 2013; Wadler, 2009; Rusty, 2010; Flanagan, 2004; Flanagan, 2015; Eubanks, 2005; Freeman, 2004) que representa al sistema de evolución numérica escrito en el sistema de ecuaciones en (5), por una parte y (7), por otra parte, pero de manera coordinada.

La forma simplificada de escribir el código sería la siguiente:

$$\begin{aligned} \omega_0 &\leftarrow \omega, \\ \omega &\leftarrow \omega - (h/2) * g / l * \sin(\theta), \\ \theta_n &\leftarrow \theta + h * \omega, \\ \theta_{n+1/2} &\leftarrow \theta + h/2 * \omega_0, \\ \omega &\leftarrow \omega_0 - h * g / l * \sin(\theta_{n+1/2}), \end{aligned} \tag{8}$$

$$\begin{aligned} \theta &\leftarrow \theta_n, \\ t &\leftarrow t + h, \end{aligned} \tag{9}$$

donde el símbolo “ \leftarrow ” quiere decir “asignación”.

Hemos tenido el cuidado de guardar valores previos de las variables, en variables temporales con subíndice cero. El subíndice “ $n+1/2$ ”, quiere decir que hemos elegido un punto medio entre “ n ” y “ $n+1$ ”, que es la esencia del método numérico del valor medio, o sea tomar valores intermedios, que aproximen mejor los valores tomados con el método directo de Euler.

Salvo la declaración de nuevas variables temporales, el código es el pasado y se muestra en la figura 3, en el editor de la “Evolución” de la parte del “Modelo” de Easy Java Simulations:

```
double omega0 = omega;
omega = omega - h/2 * g/l *Math.sin(theta);
double theta_N = theta + h*omega;
double theta_M=theta+h/2*omega0;

omega = omega0 - h * g/l * Math.sin(theta_M);

theta = theta_N;
t = t + h;
```

Realizamos algunas aclaraciones: En el código, la letra “theta” denota a la variable θ ; “theta_N” denota a la variable θ_n ; “theta_M” denota a la variable θ_{n+1} ; “omega” es la variable ω ; omega0 es la variable ω_0 y h es la variable $dt \approx \Delta t = t_n - t_{n-1}$, que es el incremento en el tiempo, para que todo el sistema del péndulo simple evolucione.

En cuanto a “g” es la aceleración de la gravedad y “l” es la longitud de la cuerda o el hilo del péndulo.

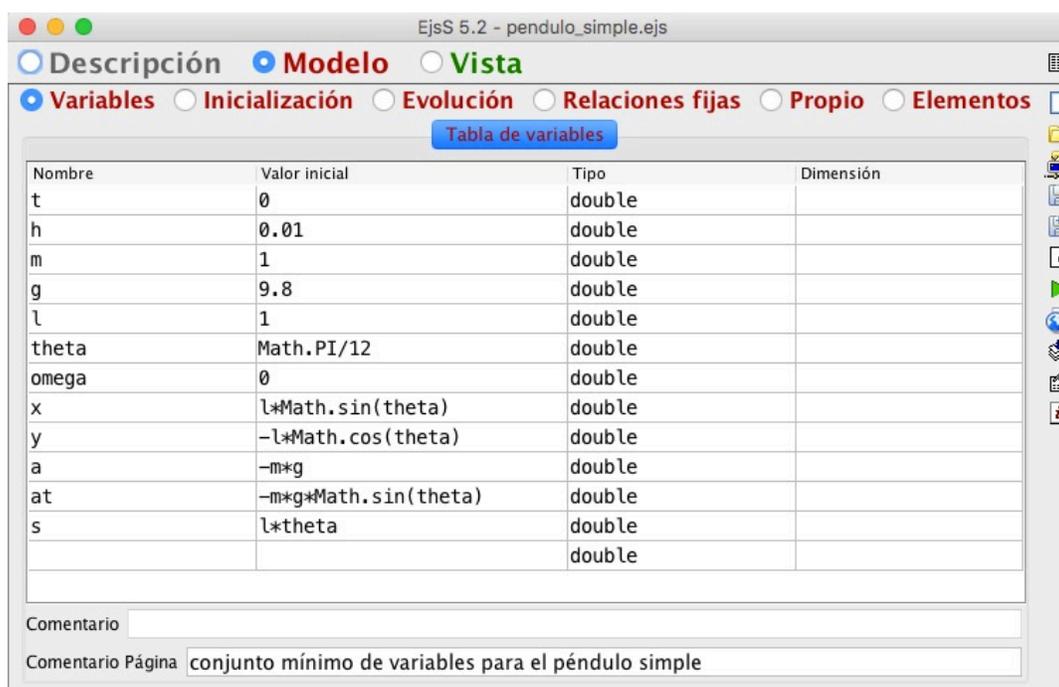


FIGURA 2. Lista de variables que toman parte de la evolución del movimiento para el péndulo simple.

Otras variables que hay que manejar son las coordenadas cartesianas “x” y “y”. Estas variables, están “ligadas” en forma permanente con las coordenadas polares l y θ adaptadas a la vertical. En la figura 4, están estas relaciones ingresadas al editor de EJS, en la subsección de EJS conocida como “Relaciones fijas”.

Después de editar la evolución del movimiento para el péndulo simple, se procede a construir una “Vista” de la simulación empleando elementos gráficos que ya provee el entorno de EJS.

XI. RELACIONES FIJAS DE LA SIMULACIÓN

En esta sección del código, disponemos de aquellas relaciones matemáticas que se mantienen, durante el transcurso de la simulación. Para construir esta sección seleccionamos el sub-radio-botón llamado “Relaciones fijas”. Luego, se edita la parte inferior de la misma con el código adecuado, como ya se mencionó antes hay varias relaciones que se mantienen con el tiempo. En la figura 4, mostramos estas relaciones fijas que son básicamente el dato de la aceleración del péndulo, las relaciones de cambio de coordenadas de polares a rectangulares, y la longitud de arco recorrida por el péndulo.

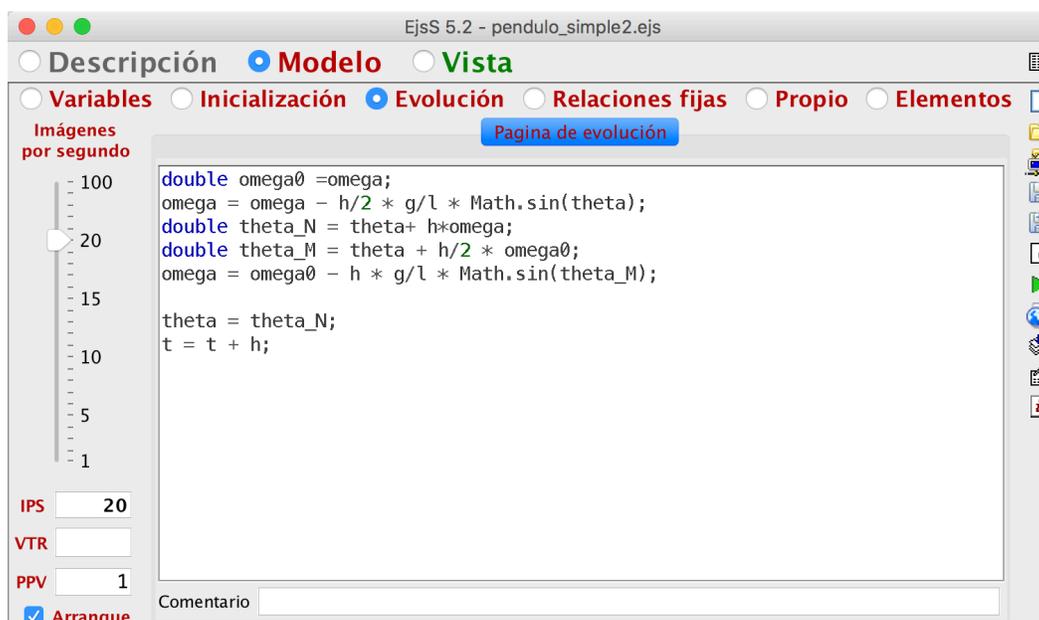


FIGURA 3. Código de evolución de movimiento escrito en el lenguaje de programación Java.

XII. CONSTRUYENDO UNA VISTA, PARA EL MOVIMIENTO DEL PÉNDULO SIMPLE

La tercera parte, en la elaboración de una simulación en EJS, lo constituye el diseño de una “vista” de la simulación que permita la interacción con el usuario del programa, que en este caso es el mismo estudiante de la clase de Física y el profesor.

Para este fin, se eligen varios elementos gráficos y se van anotando las propiedades del mismo. Estos elementos, siguen una estructura de árbol. Esto es, algunos elementos tendrán un elemento padre y uno, cero o varios elementos hijo. Habrá una raíz común, que representa en sí toda la “Vista de la simulación”. En la figura 4, mostramos un desarrollo parcial de como se van añadiendo elementos a la raíz de la vista, la cual se encuentra en la mitad izquierda de la ventana “Vista” de EJS, que se ha seleccionado activando el radio botón del mismo nombre.

Con el objeto de mostrar como se realiza una vista en la figura 8, se muestra la vista que hemos construido para la presente simulación. La Figura 9 muestra las partes de la vista, más importantes para construir la simulación. Primero hemos agregado una “Ventana” a la raíz de la vista, y tres “VentanaDialogo”. La ventana sirve para colocar al péndulo simple, las Ventanas diálogo son para colocar las gráficas de como evolucionan ciertas variables importantes con el tiempo. En la primer ventana de diálogo, colocamos la evolución de la longitud de arco que traza la pequeña masa del péndulo simple. En la segunda ventana de diálogo se muestra el espacio fase para el péndulo simple. En la tercer ventana de diálogo, se muestran la velocidad angular y la aceleración tangencial, que experimenta el pequeño cuerpo que se suspende en el péndulo. Todas estas ventanas las tomamos de la primera sección “Interfaz” que está del lado derecho de la ventana “Vista” de EJS.

En la ventana principal agregamos un “PanelDibujo”, que esencialmente es el contenedor más próximo a los elementos que constituyen al péndulo simple. Aquí colocamos un “Techo” o soporte del péndulo, un “hilo” o “cordón” y al final una “partícula”, que es el pequeño cuerpo de masa m que cuelga del hilo en el péndulo.

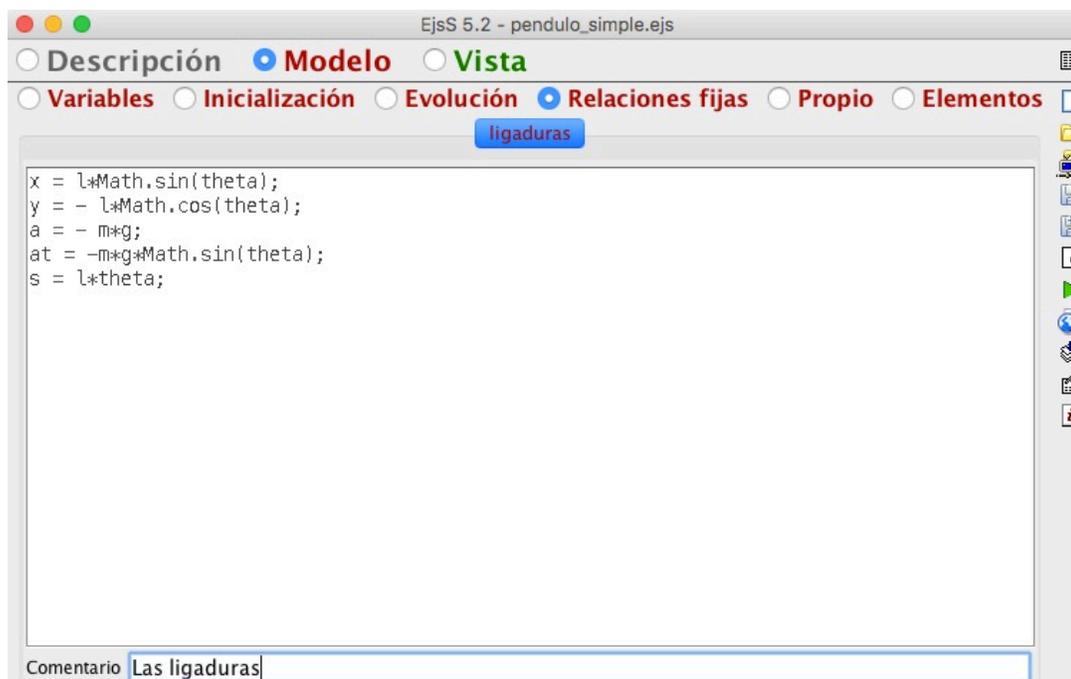


FIGURA 4. Relaciones fijas entre las coordenadas cartesianas y polares

Quizá la parte más sobresaliente de la simulación es la que representa a los elementos físicos del péndulo simple, que son el hilo y el cuerpo que cuelga, otro elemento, un poco más secundario sería el techo o soporte de donde cuelga el hilo del péndulo. Para poder editar las propiedades del elemento gráfico, que hemos nombrado “techo”, damos un click derecho de ratón a este elemento, y en seguida nos aparece un menú contextual, mostrándonos que podemos modificar las propiedades de este elemento gráfico en particular. Así seleccionamos la opción del menú “Propiedades”, y esto nos abre un cuadro de diálogo del mismo nombre, donde podemos justamente modificar las propiedades de este elemento gráfico. La figura 10, nos muestra el menú contextual, para el elemento “techo”, y la figura 11 nos muestra los valores seleccionados, del elemento techo, para que tenga la apariencia de ser el “techo” donde se apoya el hilo del péndulo.

XIII. COMO DISEÑAR EL TECHO QUE SOPORTA AL PÉNDULO

En la figura 10 se muestran las coordenadas X y Y para la posición del “techo” que soporta al péndulo, las dimensiones de su “tamaño”, y la clase de forma que adquiere, en el campo denominado “estilo”. Por ejemplo, para escoger el estilo RECTANGLE en la columna “Aspecto Gráfico” del cuadro de diálogo “Propiedades de techo (forma)”, pulsamos el primer botón que representa las propiedades de forma de “techo”, cuyo aspecto gráfico se muestra en la figura 5.



FIGURA 5. Botón para activar la forma de “techo” que en este caso sera la de un rectángulo

En la figura 6, mostramos otro cuadro de diálogo mostrándonos las opciones que tenemos para darle una forma específica a nuestro techo, y escogemos la opción “RECTANGLE”, señalándola con el ratón y oprimiendo el botón izquierdo del mismo.



FIGURA 6. Cuadro de diálogo “Elija una opción”, para escoger la forma que tendrá el techo que soporta al péndulo.

Ahora escogemos el color del techo, para ello, en la columna “aspecto gráfico”, nos vamos al campo llamado “Color de relleno”, y oprimimos el botón con el mismo aspecto que el de la figura 5, que está a la derecha del campo.

Nos saldrá el cuadro de diálogo de la figura 7, y allí escogemos el color del techo



FIGURA 7. Cuadro de diálogo “Elija una opción”, para escoger el color que tendrá el techo que soporta al péndulo.

En este caso, lo escogimos de color azul, pero puede variarse la elección.

El tipo de elemento gráfico para el “techo”, en EJS se denomina “Forma” y representa precisamente a una forma en 2 dimensiones (2D), que puede ser una elipse, una rectángulo, etc.

Las “Formas” son elementos gráficos de EJS y se pueden añadir a la “vista de la simulación”, como elementos hijos de otro elemento gráfico, en este caso, denominado “Panel de dibujo” el cual tiene el nombre “péndulo_simple” en la figura 8.

Las “Formas” se encuentran en la sección de columna derecha de la ventana “Vista” de la ventana principal de edición de EJS denominada “Elementos para la vista” y en la segunda subsección llamada “Elementos de dibujo 2D”.

La manera de construir el Panel de dibujo, es similar a como construimos el “techo”, se editan sus propiedades, siguiendo los mismos pasos, y el cuadro de diálogo llamado “Propiedades de péndulo_simple (PanelDibujo)” se muestra en la figura 8, y nos sirve de referencia para entender como se construye un panel de dibujo contenedor de los elementos gráficos del péndulo simple. Aquí se muestran las escalas de X y Y o dimensiones que va a tener el panel de dibujo.

Con esto, ya hemos aprendido a diseñar un elemento gráfico de la simulación, y hemos visto algunos de los detalles embarrasosos, con los que hay que tartar, cuando estamos construyendo la vista de la simulación. Sin embargo, todavía faltan por diseñar otros dos elementos básicos de la simulación en la ventana principal del mismo.

Estos elementos los conforman el hilo del péndulo simple, y el pequeño cuerpo que cuelga del mismo.

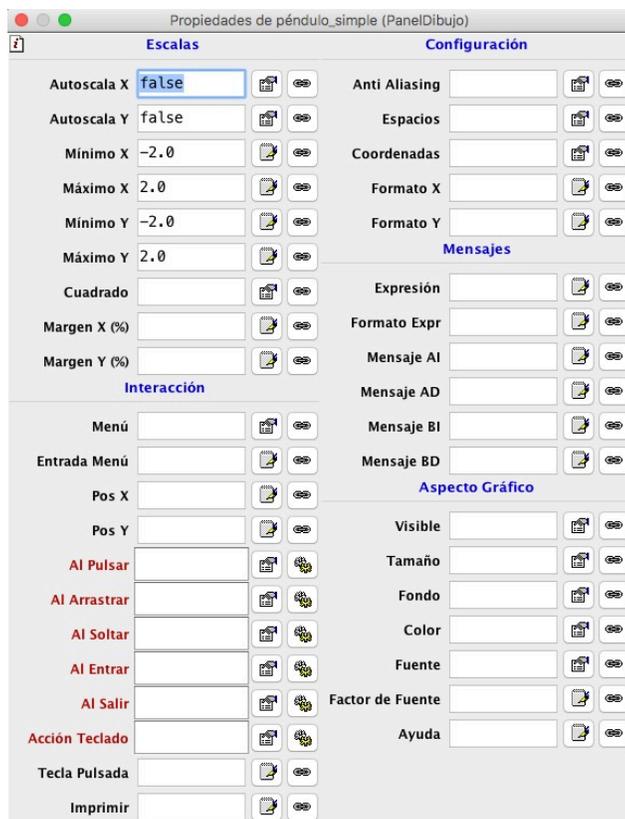


FIGURA 8. Propiedades del panel de dibujo que contiene al péndulo simple.

XIV. COMO COLOCAR EL HILO Y LA PESA

En la figura 9, se puede ver que tanto el techo como el hilo o cordon del péndulo como la particular o pesa del mismo están dentro de un mismo panel de dibujo. El techo representa el soporte del péndulo, en este ejemplo, y el hilo y la pesa son la parte principal del péndulo, que se mueven de acuerdo a como lo describe la solución a su ecuación diferencial. Por eso hay que ligar estas dos partes del péndulo simple con las variables principals que describen su movimiento, por eso es importante repasar como se pueden construir estos dos elementos para EJS.

Tanto el hilo como la pesa son objetos de dibujo 2D, y se puede colocar apartir de la sección de los elementos para la vista llamada “elementos de dibujo 2d”, que es el Segundo conjunto de objetos o de elementos de dibujo en la columna derecha de la ventana “Vista” de EJS.

Para el hilo del péndulo, escogemos una “Flecha” o “flecha 2D”, abrimos una ventana de “propiedades” como lo hicimos con el techo, nombramos a la fleche “cordon”, así que abrimos la ventana llamada “Propiedades de cordón”, que tiene tres columnas de propiedades llamadas “Posición y Tamaño”, “Visibilidad e Interacción”, y “Aspecto Gráfico”, de izquierda a derecha.

En la primer columna “Posición y Tamaño”, colocamos a las propiedades “Pos X” y “Pos Y” con los valores “0” y “0” respectivamente, esto son el origen de coordenadas esta en el punto de apoyo del hilo con el techo. Y en “Tamaño X” y “Tamaño Y” colocamos los valores de las variables “x” y “y” respectivamente. Esto es, las coordenadas cartesianas del arco descrito por la pesa, que obviamente esta sujeta al hilo en este extreme del hilo o cuerda. Esta es

una parte importante de la simulación, pues ya están en juego las coordenadas cartesianas de la variable dinámica principal de la simulación.

En la columna “Visibilidad e Interacción”, en el campo de la propiedad “Medible”, colocamos el valor “false”. Luego en la columna “Aspecto Gráfico”, y en el campo “Estilo” colocamos el valor de “SEGMENT”, en él.

Esto último es importante, pues indica que el hilo, es en efecto de la forma de un segment de línea recta. Después de esto, cerramos la ventana “Propiedades de cordón (Flecha)”, cumpliendo con la descripción y la programación de esta importante parte del péndulo simple.

En la figura 12 se muestra la ventana de propiedades del hilo o cordón del péndulo simple.

Hemos puesto esta imagen para que un posible usuario de EJS que tenga el problema de representar una simulación del péndulo simple o parecida a ella, tenga una familiaridad con la programación de una simulación en EJS.

XV. BOTONES DE CONTROL DE LA ANIMACIÓN

Con el objeto de controlar la animación del péndulo simple balanceándose de un lado a otro, y obteniendo su movimiento característico, se usa la acción de tocar y pausar la simulación con un botón de función dual. También se requiere de otro botón que tenga la propiedad de que al ser activado, regrese a la simulación desde su punto de inicio. De esta manera, es posible que el estudiante, pueda controlar el comportamiento de la simulación de manera global. Otra forma de controlarla puede consistir naturalmente, al cambiar las condiciones iniciales de la pesa que funciona como balanceo, alterando su posición y velocidad inicial, así como su masa, además se puede variar la longitud del cordón que sostiene a la pesa. De esta forma, el estudiante puede comprender el funcionamiento del péndulo simple, desde el punto de vista físico.

Claro que otra forma de controlar la simulación es a través de la forma en como esta se ejecuta. Y para esto, es que se necesitan de al menos dos botones que a la hora de oprimirlos, alteren la marcha o el paro de la simulación. O bien su retorno a las condiciones iniciales del mismo. Es decir se necesita un botón de arranque y pausa de la simulación y otro de “reset” o regreso de la simulación. También se puede aprovechar el espacio para mostrar como es que el tiempo va corriendo conforme el movimiento del péndulo va avanzando. El elemento gráfico que puede almacenar a estos otros elementos gráficos en el espacio físico donde van a aparecer dentro de la ventana de la simulación, se llama “panel de control”.

Los paneles de control, se encuentran en la columna “Elementos para la vista” en la parte superior o sección superior llamada “Interfaz”, la cual tiene cinco secciones que se eligen con cinco botones de menú que se encuentran en la parte superior de “Interfaz”, se selecciona el Segundo botón llamado “Ventanas, contenedores y paneles de dibujo”. Una vez seleccionado esto, el botón se encuentra disponible un elemento llamado “Panel: Un panel contenedor básico”. Se selecciona a este elemento del lado o columna derecha de la ventana principal “Vista” de EJS, y se coloca como elemento hijo de la ventana principal de la simulación donde se encuentra el péndulo simple.

Al nuevo panel contenedor le llamamos “botones” y ahí colocamos dos botones controladores y otro panel, el cual a su vez una “etiqueta” y un “campo numérico”, cuya función será la de mostrar como transcurre el tiempo en la simulación. En la primera sección “Interfaz” de “Elementos para la vista”, se selecciona el tercer botón del menú llamado “botones y decoración”, luego se selecciona al “BotonDosEstados”. También se puede elegir un “Botón: Botón para acciones”, que es para implementar la acción de “resetear” o regresar a la animación a su punto de condiciones iniciales.

Una vez que ambos botones están como elementos hijos del panel “botones”, hay que editar las propiedades de cada botón, a fin de que realicen la función para la cual están pensados

XVI. GRÁFICA DEL DESPLAZAMIENTO DE ARCO DEL PÉNDULO SIMPLE

En la parte de “Interfaz”, se escoge una VentanaDiálogo de la sección “Ventanas, contenedores y paneles de dibujo”, se coloca en la raíz “Vista de la Simulación”, y le llamamos “Longitud_de_arco”. Como hijos de este elemento ponemos un “PanelConEjes” (al que llamamos “Longitud_de_arco_vs_t”) y un “Panel” (panel contenedor básico), el cual luego explicaremos. El panel con Ejes sirve para colocar la curva principal, de las gráficas que acompañan a la simulación. Aquí es donde se grafica el “desplazamiento de arco” del péndulo simple en función del tiempo. Abrimos la ventana “Propiedades de Panel (PanelConEjes)”. Y rellenamos Autoscala X con true, Autoscala Y con false, Mínimo X con -0.3, Máximo X 1.0, Mínimo Y -0.3, Máximo Y con 0.3, cuadrado con false, esto es en la columna de

Escalas, que la intención es que la gráfica sea visible. En Decoración y Título se pone un título apropiado para la gráfica como “Desplazamiento”, en la columna Configuración y campo “Fixed Gutters” podemos poner el valor “true”. Todo esto es para predisponer de una gráfica con el desplazamiento de arco de la masa o partícula del péndulo simple, contra el tiempo. Para esto colocamos ahora el hijo “rastros: Un rastro 2D”, de este panel, lo llamamos por ejemplo oscilaciones, y en sus Propiedades colocamos como “Entrada X” al tiempo t y como “Entrada Y a la longitud de arco s ”. De esta forma podemos graficar la función principal de la simulación que es la longitud de arco recorrida por la pesa en función del tiempo o $s=s(t)$. Esta sería casi por completo el objetivo principal de la simulación: Presentar la variable principal del fenómeno y de la ecuación diferencial del péndulo simple s como función de su variable independiente, que es el tiempo t . Se pueden añadir otras gráficas que van a proveernos de información adicional, por ejemplo, se puede sugerir realizar una gráfica para el espacio fase del péndulo simple. El espacio fase no es más que una gráfica donde se grafica la velocidad tangencial de la partícula del péndulo simple, contra su desplazamiento. Para ángulos pequeños esta gráfica es aproximadamente de la forma de una elipse. En este ejemplo, lo podemos poner como una información adicional muy bonita que complementa al comportamiento de la longitud de arco que se desplaza la partícula, en función del tiempo.

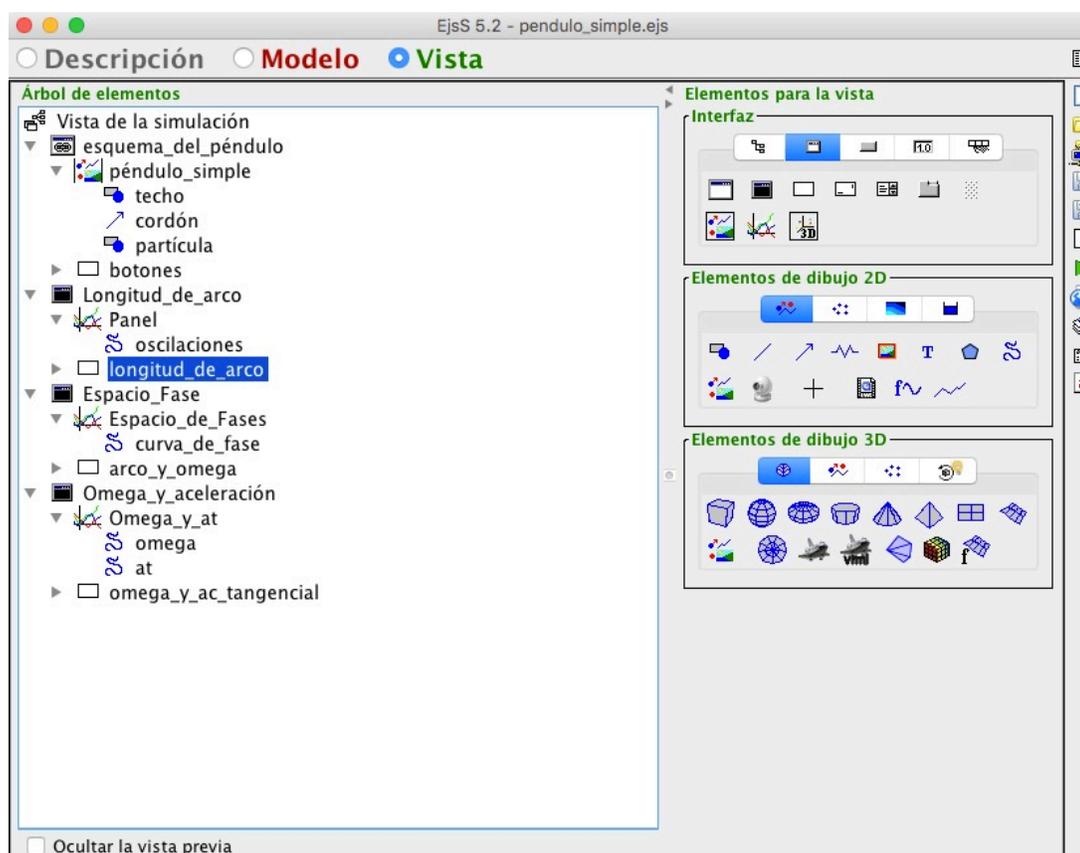


FIGURA 9. Ventana “Vista” de la simulación elegida a partir del radio botón “Vista” del menú superior de EJS. Se muestra un desarrollo parcial de los elementos de la simulación. En la parte izquierda de la ventana, está la “vista” construida de la simulación, y en la parte derecha se muestran tres paneles, que tienen varios elementos gráficos, con los cuales se puede ir construyendo la vista, añadiendoselas al elemento raíz “Vista de la simulación”, estos elementos se “seleccionan con el ratón en la parte derecha y luego se hace un click a otro elemento del lado izquierdo, añadiendolo como elemento hijo del mismo.

Como parte final de la ventana “Longitud de arco (ventana de EjsS), ponemos un elemento “Panel: Un panel contenedor básico”, como hijo de la VentanaDiálogo “Longitud de arco”, y como hijo de este Panel ponemos una “etiqueta: una etiqueta decorativa”, para poner un texto informativo diciendo cuales son las cantidades graficadas, pues

en la dirección horizontal estará el tiempo t , y en la dirección vertical estará la longitud de arco recorrida por la pesa del péndulo simple s . De esta manera un usuario sabrá las cantidades que se están graficando en esta ventana.

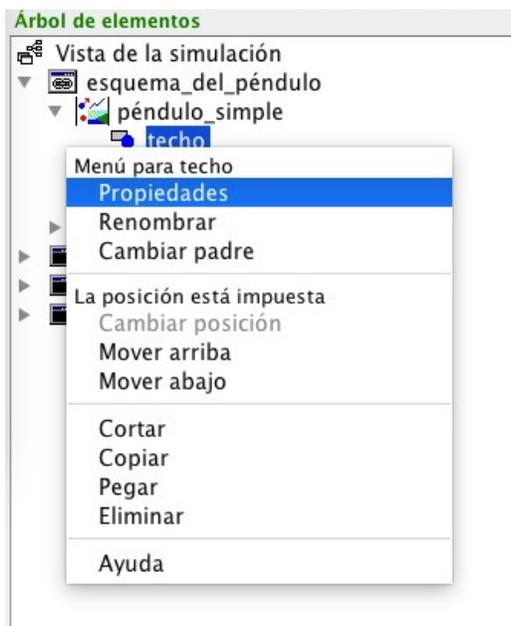


FIGURA 10. Menú contextual para el elemento gráfico denominado “techo”, para editar sus propiedades elegimos la opción del mismo nombre.

XVII. GRÁFICA DEL ESPACIO FASE

Para que la animación se vea más completa, podemos opcionalmente, añadir una gráfica de espacio fase para el péndulo simple. Así que añadimos una *VentanaDiálogo* como hijo de la raíz *Vista de la simulación* y la llamamos “Espacio_Fase”, y añadimos un *PanelConEjes* como hijo de la *VentanaDiálogo* al cual llamamos con el nombre de “Espacio_de_Fases”, para la gráfica añadimos un *Rastro: Un rastro 2D* y la llamamos, por ejemplo *curva_de_fase*. Al editar sus propiedades colocamos como “Entrada X” a la variable s , y como “Entrada Y” a $l*\omega$, otras propiedades son Max Puntos de 190, No Repetir como true, Conectar como true, Color de Línea como BLUE, y Ancho Línea de 0.4. La gráfica del espacio fase tiene la forma aproximada de una elipse, como lo podemos comprobar a la hora de correr la simulación.

A manera de etiqueta informativa añadimos un Panel de hijo para la *VentanaDiálogo*, y adentro ponemos una Etiqueta: Una etiqueta decorativa, y a la hora de editar sus propiedades ponemos como texto “ $s \leftarrow$ horizontal” y “ $l*\omega \leftarrow$ vertical”, indicando que es lo que se grafica en el eje horizontal y que se grafica como eje vertical. En un espacio fase, se toman las cantidades de velocidad tangencial y la longitud de arco, como variables a graficar. La gráfica resultante es aproximadamente una elipse.

XVIII. GRÁFICA DE OMEGA Y DE LA ACELARACIÓN TANGENCIAL

Con el objeto de programar una animación que arroje más datos, podemos graficar la velocidad angular omega y la aceleración tangencial en una gráfica aparte, usando colores distintos para cada gráfica. Ambas cantidades se grafican contra el tiempo, y se obtiene gráficas descriptivas del fenómeno de movimiento para el péndulo simple.

Añadimos entonces otra *VentanaDiálogo* como hijo de la raíz *Vista de la simulación* y esta vez la llamamos Omega y Aceleración, para tener ejes de coordenadas añadimos como hijo de esta *VentanaDiálogo* un *PanelConEjes* y anotamos en sus propiedades el rango de valores que tendrán tanto omega (velocidad angular) como la aceleración tangencial.

Las gráficas en sí las añadimos como dos *Rastro: Un rastro 2D*, que es el nombre del elemento gráfico que se usa para colocar las gráficas de omega y la aceleración tangencial de la pesa en función del tiempo. A la primer gráfica la llamamos “omega” y a la segunda “at”, lo importante aquí es que identifiquemos cada gráfica y no el nombre que usemos para designar a cada una.

La parte más importante de las propiedades que se editan para la gráfica “omega”, son “Entrada X” en la cual estará ligada la variable t , y la “Entrada Y”, en la cual se ligará a ala variable ω . Las demás propiedades como el color de la línea se pueden ajustar al gusto.

La parte más importante que se edita de las propiedades para la gráfica “at” son “Entrada X” en la cual se liga a la variable independiente del tiempo t , y la “Entrada Y” la cual estará ligada a ala variable at , que es la aceleración tangencial de la pesa.



FIGURA 11. Cuadro de diálogo “Propiedades”, para el elemento “techo” de la simulación. Se muestra la posición del techo, sus dimensiones, su forma de “rectangle” y su color de relleno.



FIGURA 12. Cuadro de diálogo “Propiedades”, para el elemento “cordón” de la simulación. Se muestra la posición del hilo, sus dimensiones, su forma de “segmento”.

XIX. GUARDAR LA SIMULACIÓN

Con el objeto de guardar nuestra simulación, en el menú de EJS nos vamos a “File” y luego a “Guardar”, en una Máquina Apple esto es equivalente a presionar las teclas command-S. También en el menú del lado derecho de EJS hay una figura con forma de diskette, donde se puede oprimir para guardar nuestro trabajo. También hay otra opción que nos permite empaquetar nuestra simulación. Para mejorar el control del programa mientras escribimos nuestra simulación también en el menú de la derecha hay un botón con forma de triángulo verde acostado, que nos permite “ejecutar” la simulación para ir revisando nuestros avances. También hay botones para crear una simulación nueva y para “abrir” una simulación existente. Es importante que nosotros nos familiaricemos con todos estos botones, pues son fundamentales para el manejo del programa EJS.

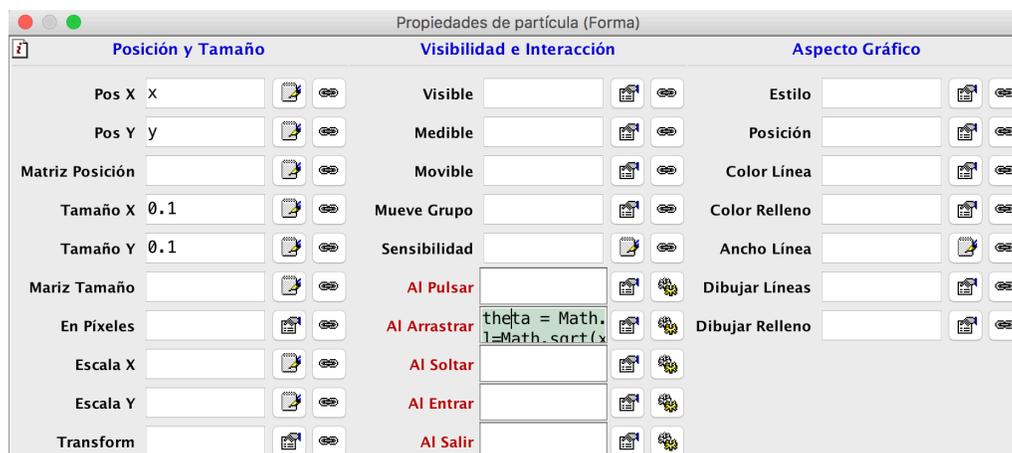


FIGURA 13. Cuadro de diálogo “Propiedades”, para el elemento “partícula” de la simulación. Se muestra la posición de la partícula, sus dimensiones, su forma de “segmento”. Y la acción o código que se ejecutará, al arrastrar el cuerpo o particular del péndulo simple.

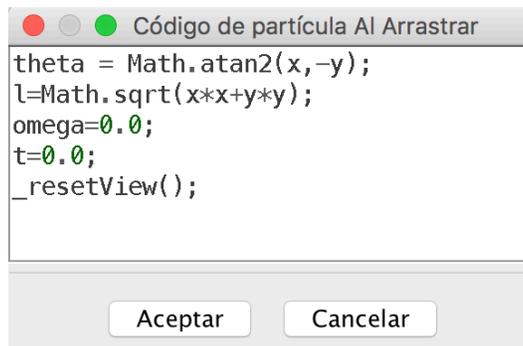


FIGURA 14. Cuadro de diálogo “Código de particular Al Arrastrar”.

XX. EJECUCIÓN DE LA SIMULACIÓN

Como ya hemos mencionado con anterioridad, la simulación se puede ejecutar con un botón del menú derecho de EJS con forma de triángulo acostado de color verde.

Si hemos escrito nuestra simulación con el debido cuidado deberíamos de observar que se abren cuatro ventanas. En una ventana se encuentra la simulación del péndulo simple, propiamente dicha, con botón de play y pause, y botón de reset. Además hay otras tres ventanas con las gráficas de las variables descritas por la simulación.

En la clase de Física es importante que los estudiantes corran la simulación para que puedan comprobar su aprendizaje sobre el péndulo simple. Además se sugiere que desarrollen el código con ayuda del profesor.

Aquí se muestra como debería de verse una simulación bien escrita.

Es responsabilidad del profesor y de los estudiantes que la simulación tenga una buena realización y se desempeñe correctamente. Además que es importante que el profesor implemente los pasos del ciclo PODS para que el estudiante se vuelva dinámico y su aprendizaje sea efectivo y bueno.

En la figura 15 se muestra un ejemplo de la simulación del movimiento para el péndulo simple. La simulación se preparó para una máquina Apple pero puede implementarse en una máquina de fábrica con Windows, o una máquina armada con Windows, Linux, FreeBSD, u otro sistema operativo que pueda soportar correr programas en el lenguaje de programación Java. Como este lenguaje esta bien difundido hoy en día, es prácticamente posible realizar la simulación casi en cualquier sistema operativo conocido.

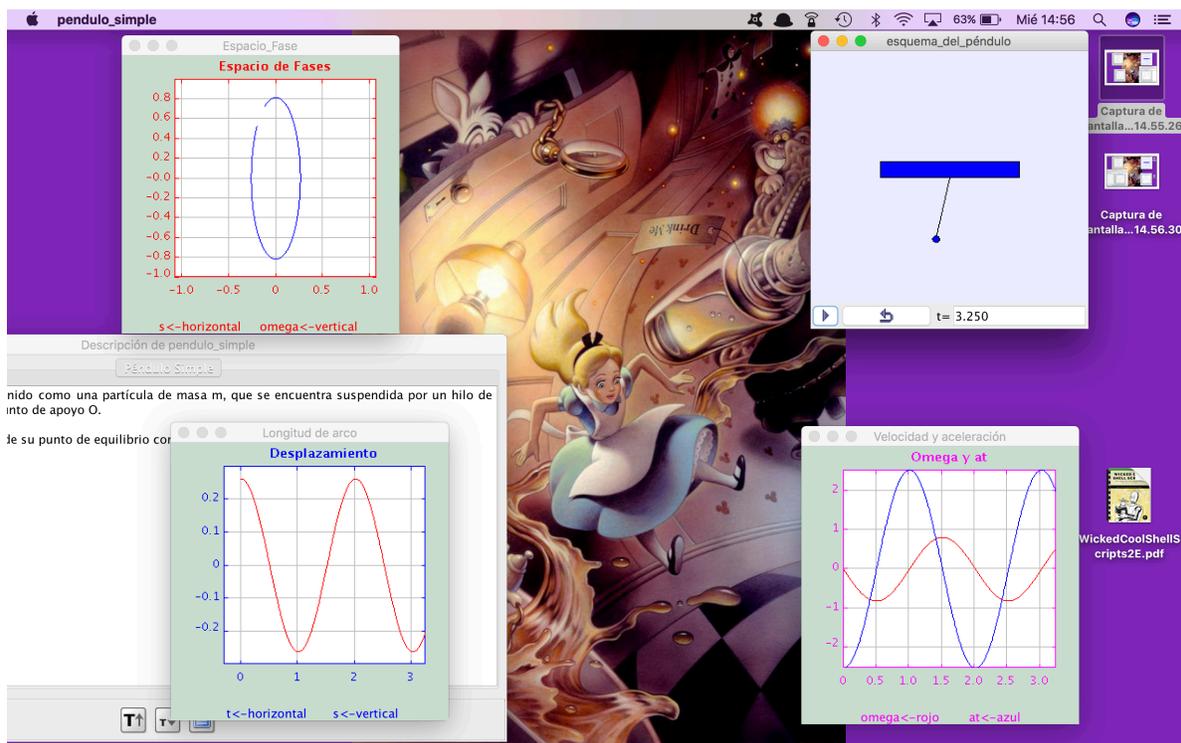


FIGURA 15. Ilustración de cómo se ejecuta la simulación creada en EJS del péndulo simple, para un sistema operativo OS. Se aprecian las ventanas que despliegan varias gráficas y una ventana donde se simula el movimiento del péndulo con botones de control de la simulación. Entre las gráficas están las del desplazamiento de arco, el espacio fase, y la de la velocidad angular y la aceleración tangencial de la pesa. También una ventana donde hay una breve descripción textual del péndulo simple, que nosotros mismos podemos agregar para complementar la información teórica dada al estudiante.

XXI. CONCLUSIONES

En este trabajo se ha mostrado como sugerencia el implementar una estrategia didáctica basada en el ciclo PODS y complementarla con un trabajo de programación sencilla en EJS, la simulación formada es la del movimiento de un péndulo simple. El alumno, puede aprender tanto la naturaleza del movimiento descrita por la ecuaciones diferenciales, como la técnica matemática numérica que hace posible aproximar la solución a la simulación del movimiento. Se espera que una propuesta así, pueda ayudar a mejorar la calidad de aprendizaje que tienen los estudiantes de nivel bachiller o posterior, y que ellos aprovechen mejor su tiempo, y puedan apreciar por si mismos la forma en como la

ciencia describe el movimiento del péndulo simple. Este es ciertamente una propuesta sencilla que se puede llevar al salón de clases, si se tienen los recursos necesarios como lo puede ser el equipo de cómputo adecuado. Hay muchas posibilidades de que la escuela disponga de el equipo de cómputo adecuado, o de que el propio estudiante, tenga un equipo propio, pues la tecnología de las computadoras ya sea extendido bastante a muchos países en las últimas décadas. Por lo tanto, podemos concluir que esta propuesta puede ser efectuada por alguna institución de presupuesto mediano, y da la ventaja de que propone que sea el mismo estudiante quien participe en su propia formación, teniendo en cuenta que es él quien aprende, esta propuesta se ve que es bastante prometedora, al menos ara el tema del péndulo simple. Asimismo se ha descrito en detalle, como manejar el paquete de simulación EJS de Francisco Esquembre para lograr un buen resultado en el programa de la simulación, el cual se puede ajustar de acuerdo a las preferencias y necesidades individuales que se tenga en la clase. Luego, esperamos que este sencillo ejemplo clarifique el uso y las ventajas que tienen el uso de tecnología moderna, como auxiliar en la educación.

XXII. AGRADECIMIENTOS

Los autores quieren expresar su agradecimiento al CONACYT o Consejo Nacional de Ciencia y Tecnología de México, y también al Sistema Nacional de Investigadores (SNI) del mismo CONACYT, que lo han apoyado con la realización de este trabajo. También se quiere dar el agradecimiento a la COFAA del Instituto Politécnico Nacional de la Ciudad de México por la ayuda recibida. Y al proyecto SIP (Secretaría de Investigación y Posgrado) del IPN no. 20161695. *Uso de prototipos experimentales para la enseñanza del movimiento rotacional en ingeniería – Diseño del test de evaluación.*

REFERENCIAS

- Bates, B., Sierra, K. (2009). *Head First Java*, California, USA: 2a Ed. O'Reilly & Associates Inc.
- Daniels, H. (2011). *Vygotsky and Pedagogy*. U.K.: 1ª Ed. Routledge.
- Daniels, H. (2008). *Vygotsky and Research*. U.K.: 1ª Ed. Routledge.
- Daniels, H. (2012). *Vygotsky and Sociology*. U.K.: 1ª Ed. Routledge.
- Esquembre, F. (2005). *Creación de Simulaciones Interactivas en Java. Aplicación a la Enseñanza de la Física*. Madrid, España: Pearson Prentice Hall.
- Esquembre, F. *Easy Java Simulations*. Recuperado de <<http://fem.um.es/Ejs/>>, consultado el 02 de Enero de 2016.
- Eubanks, B. D. (2005) *Wicked Cool Java*. San Francisco: No Starch Press.
- Flanagan, D. (2015). *Java in a Nutshell, A Desktop Quick Reference*. USA: 5ª Ed. O'Reilly Media.
- Flanagan, D. (2004). *Java Examples in a Nutshell, A tutorial Companion to Java in a Nutshell*. Massachusetts, USA: O'Reilly & Associates Inc.
- Freeman, Eric, Freeman, Elisabeth, Sierra, K., Bates, B., (2004). *Head First Design Patterns*. California, USA: O'Reilly & Associates Inc.
- Goldstein, H. (1980). *Classical Mechanics*. New York: 2a Ed. Addison-Wesley, NewYork.
- Google. (2016). *Programación Java. Tutorial Java. Aprende a programar con Java desde cero*. Recuperado de <http://puntocomnoesunlenguaje.blogspot.mx/2012/04/tipos-de-datos-java.html>, consultado el 02 de Enero de 2016.

Niemeyer, P., Leuck, D. (2013). *Learning Java*. California, USA: 4a Ed. O'Reilly Media.

Oracle Technology Network. (2016). Recuperado de <<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>>, consultado el 02 de Enero de 2016.

Piaget, J., Inhelder, B. (1969). *The Psychology of the Child*. New York: 2ª Ed. Basic Books.

Piaget, J. (2003). *The Psychology of Intelligence*. U.K.: 2ª ed. Routledge.

Rusty, E. (2010). *Java I/O*. California, USA: 2ª Ed. O'Reilly & Associates Inc.

Sokoloff, D. R., Thornton, R. K. (2006). *Interactive Lecture Demonstrations, Active Learning in Introductory Physics, The Physics Suite*. USA: 1ª Ed. John Wiley & Sons, Inc.

Sokoloff, D. R., Thornton, R. K. (2011a). Laws, P. W., *Realtime Physics Active Learning Laboratories, Module 1: Mechanics*, USA: 3ª Ed. John Wiley & Sons, Inc., USA.

Sokoloff, D. R. (2011b). *Realtime Physics Active Learning Laboratories, Module 2: Heat and Thermodynamics*. USA: 2ª Ed., John Wiley & Sons, Inc.

Sokoloff, D. R., Laws, P. W., Thornton, R. K. (2012a). *Realtime Physics Active Learning Laboratories, Module 3: Electric Circuits*. USA: John Wiley & Sons, Inc.

Sokoloff, D. R., Laws, P. W., Thornton, R. K. (2012b). *Realtime Physics Active Learning Laboratories, Module 4: Light and Optics*, USA: John Wiley & Sons, Inc.

Wadler, P., Naftalin, M. (2009). *Java Generics and Collections*. California, USA: 1ª Ed., O'Reilly & Associates Inc.