

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**

**DEPARTAMENTO DE ELECTRÓNICA**

## Tarea N°2 ELO329

Documentación Simulación Gráfica de Bolas,  
Puntos fijos, y Resortes como Objetos de Software

Desarrolladores:

Marcela Polanco Rol: 2821058-2

Bernardo Farias Rol: 2904681-6

Francisco Garcia Rol: 2821036-1

## Índice

1.-Introducción.....	pág. 3
2.-Elementos con los que trabaja el simulador.....	pág. 3
3.-Requisitos del Sistema.....	pág. 3
4.-Descripción de la estructura del Software.....	pág. 4
5.- Resultados Obtenidos.....	pág. 5
6.- Dificultades presentadas durante el desarrollo.....	pág. 7

## 1.-Introducción

Este documento describe como se realizo la implementación de un Laboratorio de Física en el cual se puede observar el comportamiento que tendría una configuración de elementos como bolas y resortes en 1 dimensión, todo esto gracias a la modelación del comportamiento que tendrían estos elementos en el mundo real mediante las leyes de la naturaleza.

## 2.-Elementos con los que trabaja el simulador

Dentro de los elementos que se encuentran disponibles para utilizar en la simulación están:

**Ball:** Se pueden crear bolas de un radio variable definido por el usuario. Cabe señalar que la simulación toma en consideración las colisiones que puedan existir entre una o *mas* bolas y no se considera efecto de alguna fuerza gravitatoria ni momentum angular de las bolas, además las colisiones son 100% elásticas, por lo que de darse el caso están afectaran al resultado esperado.

**Spring:** Se pueden crear resortes de una constante de rigidez y largo natural dados, ambos con la posibilidad de ser definidos libremente por el usuario, los cuales no tendrán masa y cumplirán con la ley física que los rige en la realidad (Ley de Hooke), se supone también que los mismos no salen del régimen lineal para cualquier largo dado.

**FixedHook:** Se pueden añadir puntos fijos para poder atar a estos un resorte si es necesario.

## 3.-Requisitos del Sistema:

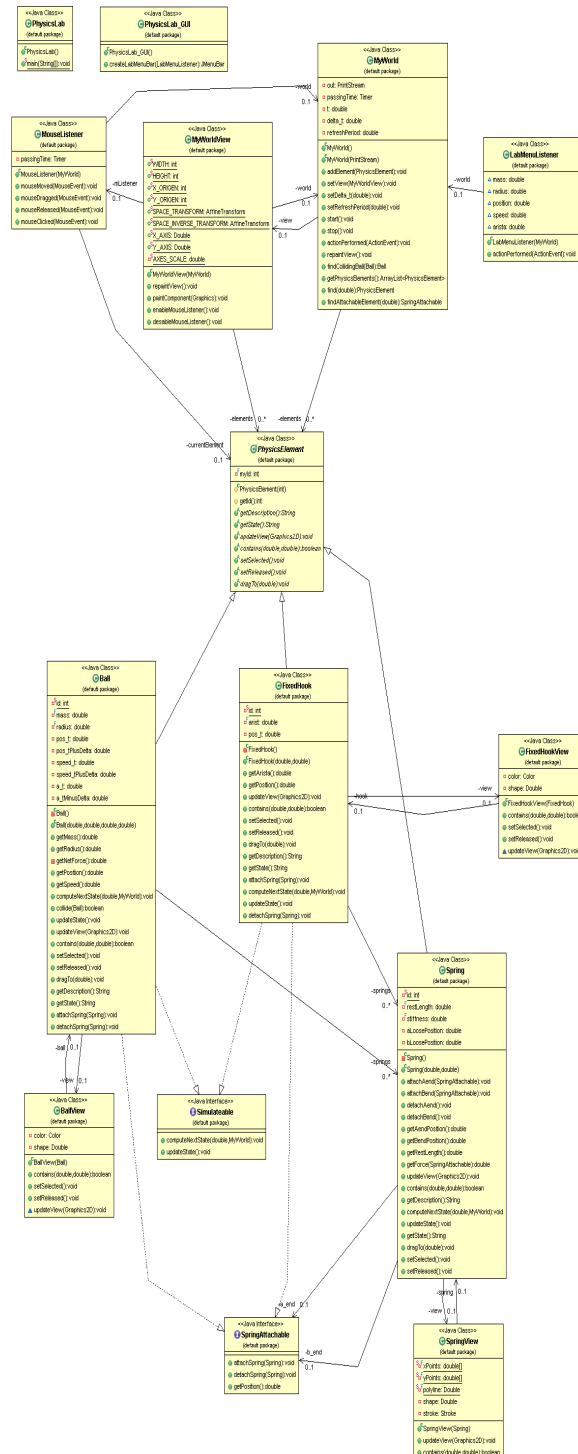
Esta aplicación debido a que esta desarrollada en el lenguaje Java puede ser corrida en cualquier sistema operativo ya sea Windows, Linux, MacOS , con la condición de tener instalado en este mismo la maquina virtual java la cual se encargara de interpretar el bytecode generado durante la compilación. Esta característica hace de este software un sistema altamente portable.

Actualizaciones: Debido a que se a decidido respetar el principio de encapsulación en este proyecto le dará la ventaja de poder ampliar sus capacidades de manera fácil y rápida en el caso de requerirlo.

#### 4.-Descripción de la estructura del Software

A continuación se presenta un diagrama UML estándar, en el cual se puede apreciar las relaciones de herencia entre clases , sus respectivos métodos implementados hasta el momento. Se puede ver aquí nuevamente como el principio de encapsulación esta presente.

Diagramas UML de las clases utilizadas en la construcción del simulador:



## 5.- Resultados Obtenidos

En este ítem se presenta lo obtenido en las simulaciones de experimentos reales en condiciones ideales.

### Primera etapa

La idea de esta etapa fue poder aprender lo que es interfaz gráfica, por lo que se completo el código de partida presentado por el profesor en la pagina del ramo, mostrando a través del interfaz un choque elástico entre 2 bolas. Dando como resultado de lo realizado la Figura 1.

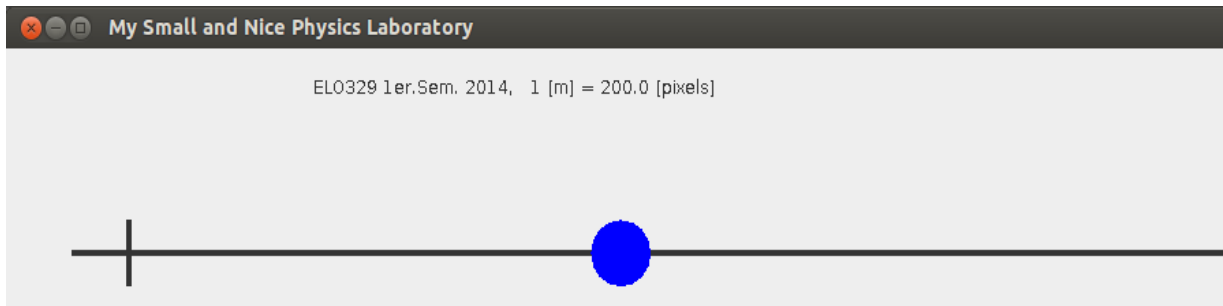


Figura 1: Interfaz para Etapa 1, choque elástico de bolas

### Segunda etapa

Luego, de ya estar familiarizados con la interfaz gráfica, se empezó a trabajar sobre ella incorporando a la etapa anterior un menú. En este caso se utiliza insertar escenario desde el menú "Configuration", creando el mismo escenario de la etapa 1. Para que parta la simulación es necesario ingresar los datos en MyWorld->Simulator, después apretar Start que se encuentra en la misma pestaña, si se desea detener solo debe apretar Stop. Lo obtenido se puede ver en la Figura 2.

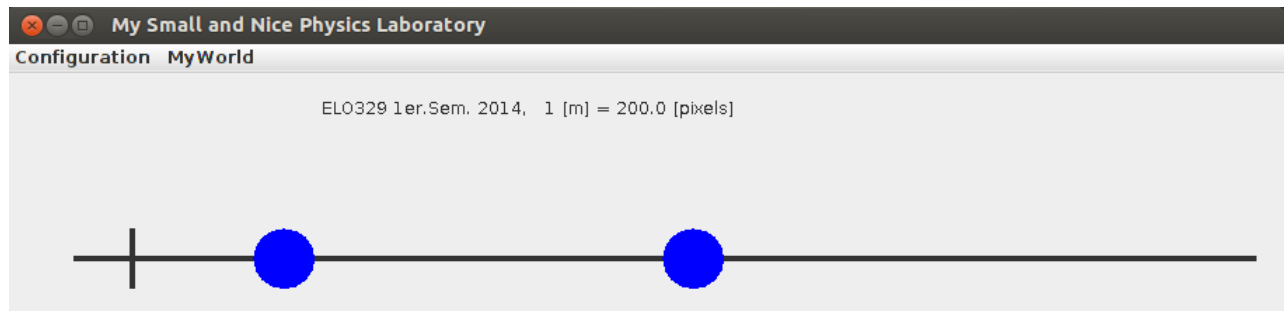


Figura 2: Etapa 1 con interfaz gráfica para insertar un escenario predefinido y con las opciones del menú MyWorld funcionando

### Tercera etapa

Trabajando sobre la etapa 2, ahora se incorporan las clases resorte y punto de apoyo fijo (Spring y FixedHook). Ahora cuando se carga “My escenario”, este muestra un bloque-resorte-bola, sigue los mismo pasos que la etapa 2 para realizar la simulación. De lo desarrollado, se puede observar en la Figura 3.

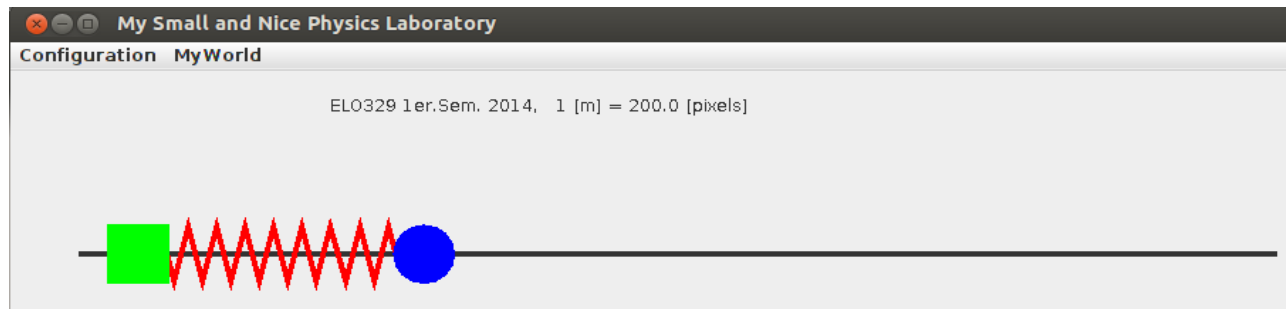


Figura 3: Se extiende Etapa 2 para permiti incorporación del escenario mostrado, es decir agrega Spring y FixedHook con su vistas

### Cuarta etapa

En esta etapa, ya se tienen todos los elementos gráficos de la etapa anterior. Ahora, se tiene la opción de insertar los elementos que el usuario desee colocándolos en el orden que estime para la realización de la simulación. Un ejemplo se puede apreciar en la Figura 4.

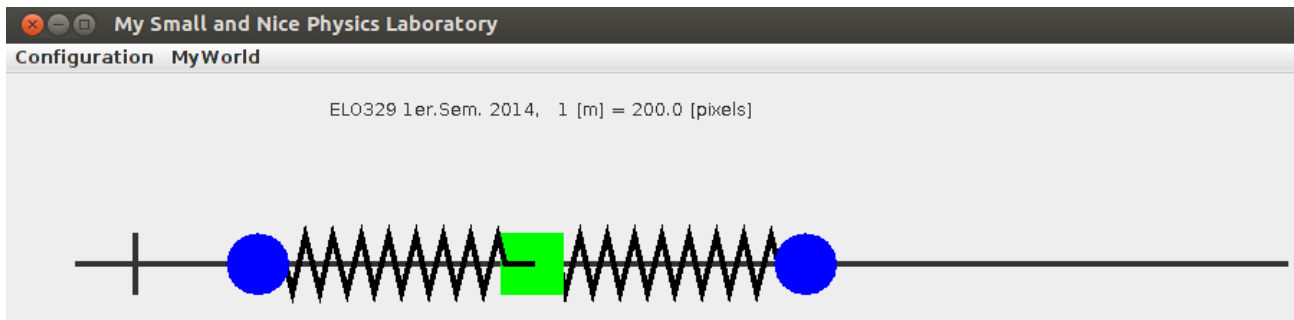


Figura 4: Extensión de la Etapa 3 para permitir edición de las configuraciones usando el mouse

## 6.- Dificultades presentadas durante el desarrollo

- Para la realización de todas las etapas, fue necesario realizar un estudio en profundidad de los contenidos vistos en clases, además de buscar códigos por internet para ayudar en la realización de la interfaz gráfica, dado a que habían muchos métodos que no supimos como funcionaban. Un ejemplo que ayudo en el desarrollo fue:

<http://www.jc-mouse.net/java/animacion-basica-en-java2d-pelota-de-futbol>

- Otro problema que se presento, fue la asociación e JMenuBar con LabMenuListener, dado a que se tienen para agregar objetos, no se tenia claridad de como se agregaba cada elemento asociado entre si.

- En la ultima etapa, para cuando se incorporaban los elementos en la interfaz, nos demoramos en ver los errores que se tenían dado a que estos no se unían entre si.