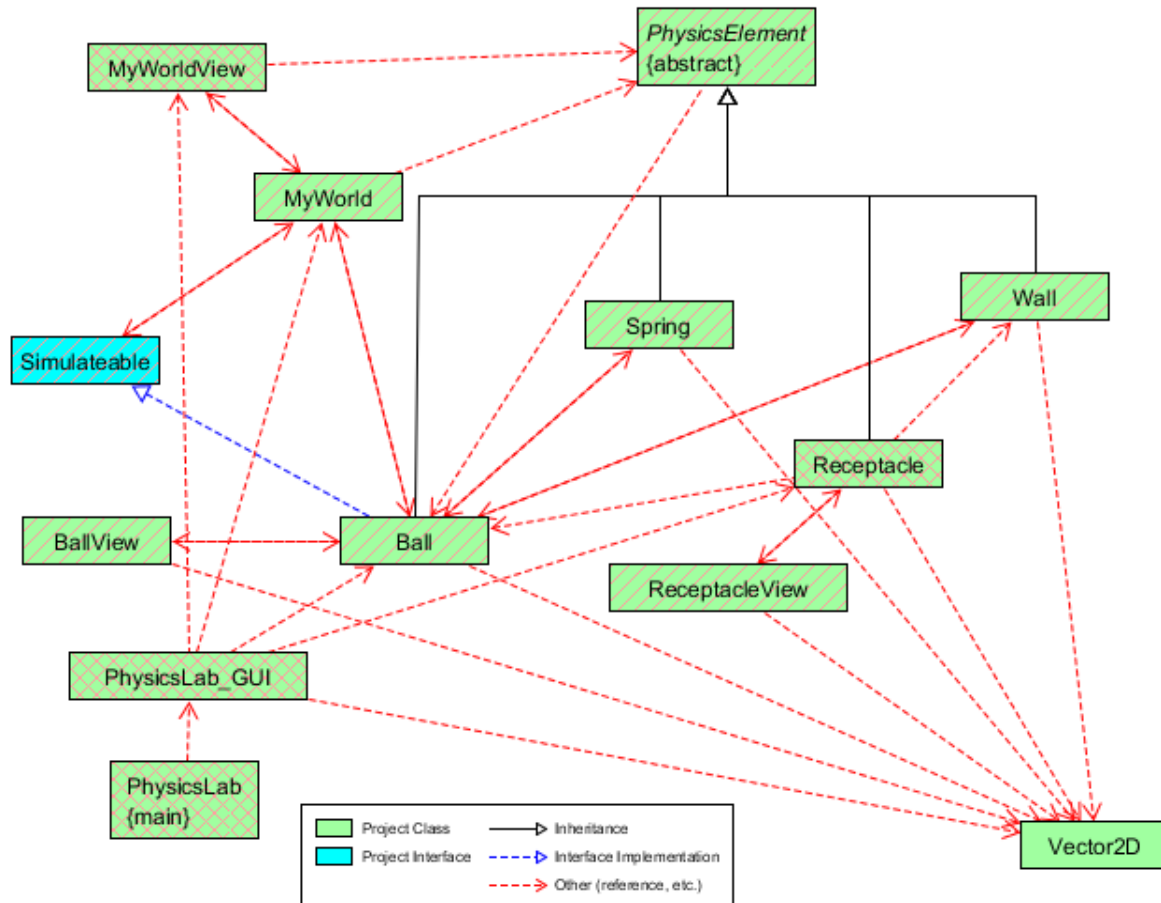


## Etapa 1: Ambiente gráfico para tarea 1, receptáculos y bola.

En esta primera etapa se implementa un ambiente gráfico al código entregado, que corresponde a una solución parcial a la tarea1.



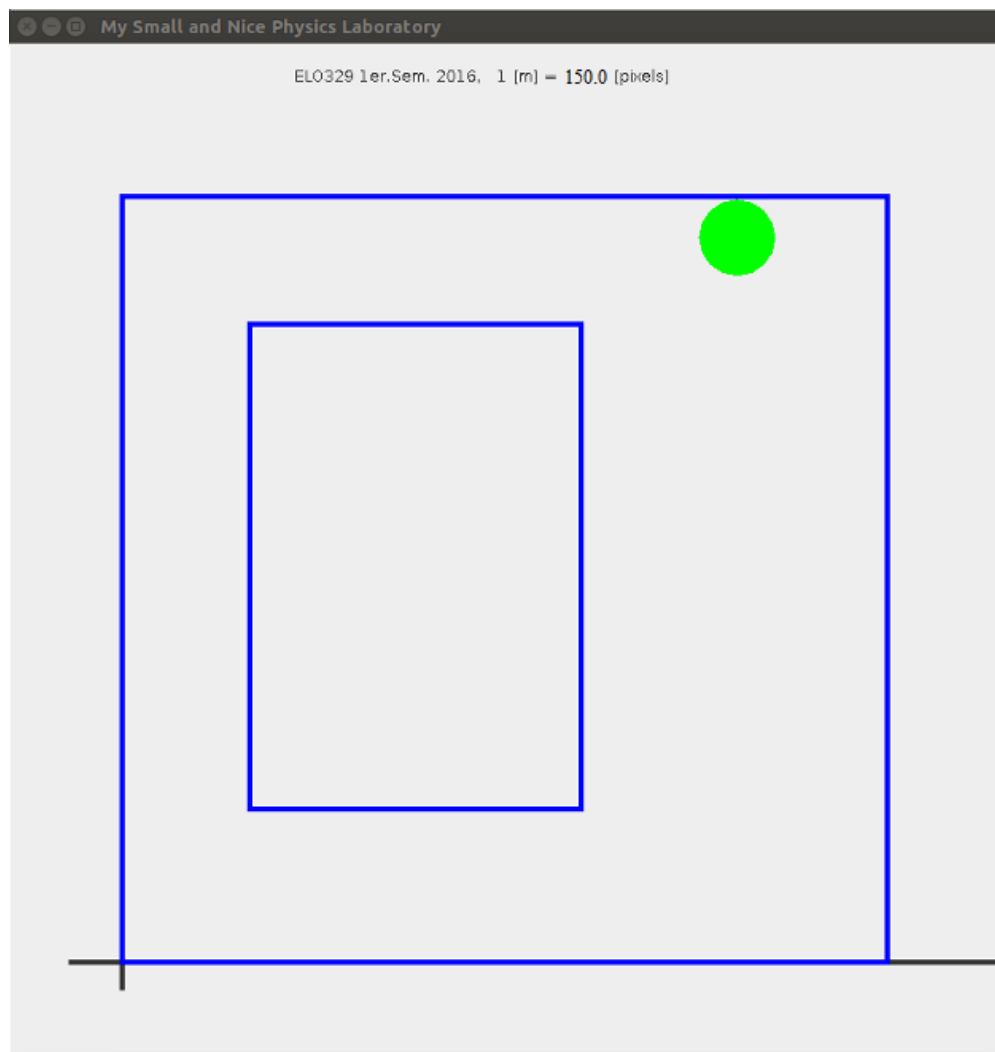
**Figura 1.** Diagrama de alto nivel para etapa 1.

La figura 1 muestra el diagrama con las clases implementadas.

Al ejecutarse el programa, se genera una configuración como la mostrada en la figura 2.

### Dificultades y/o observaciones:

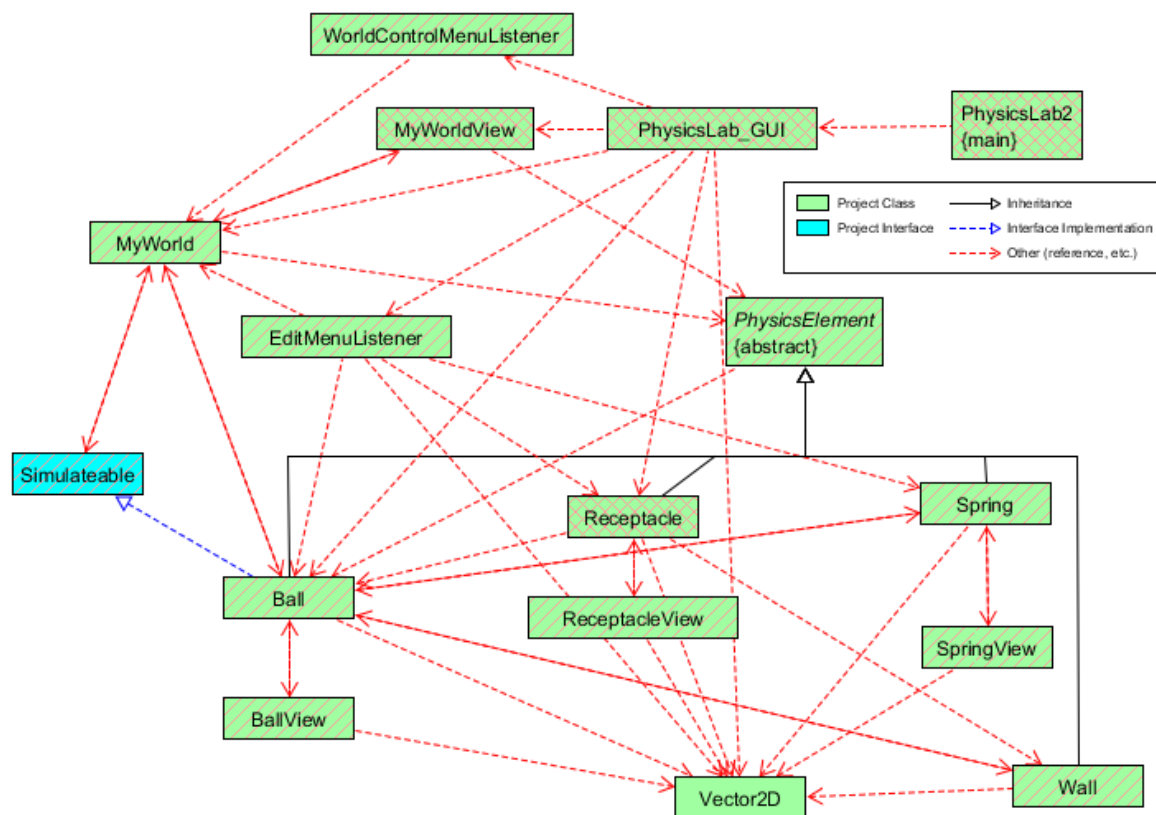
En esta etapa no se tuvieron mayores dificultades a las de comprender el código entregado.



**Figura 2.** Escenario resultante de la etapa 1.

## Etapa 2: Incorporación de menús al ambiente.

En esta etapa se incorpora la barra de menú al ambiente con dos menús, Edit y MyWorld. Con la opción Edit se pueden insertar elementos físicos, bolas, receptáculos y springs (Sólo se podrán ver un solo receptáculo y spring, ya que los elementos tienen la misma posición inicial y no se pueden mover). Con la opción MyWorld se puede iniciar y detener la simulación, y modificar los parámetros de simulación.



**Figura 3.** Diagrama de alto nivel para etapa 2.

La figura 3 muestra el diagrama con las clases implementadas.

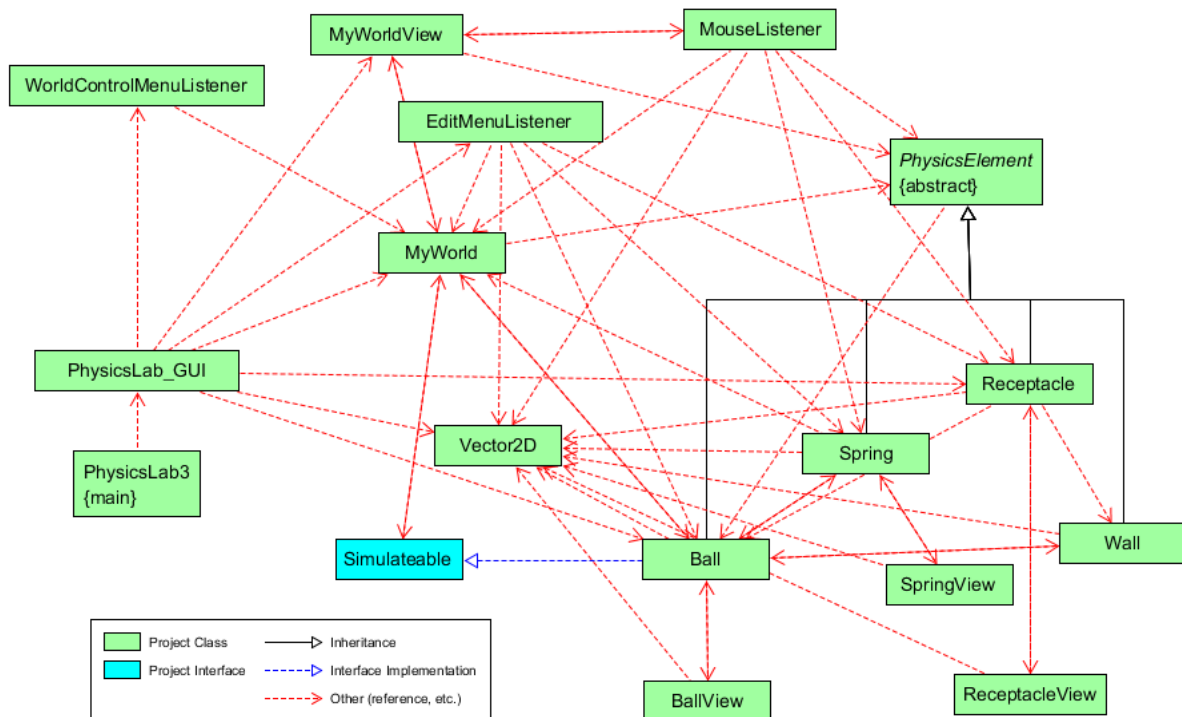
### Dificultades y/o observaciones:

En esta etapa no se tuvieron mayores dificultades a las de comprender el código entregado, y luego proceder a completarlo.

Una dificultad fue el cómo mostrar el resorte por pantalla, ya que la posición de este estaba definida por las bolas que tuvieran aferradas a este. La solución fue que, al pedir las posiciones de las bolas, si estas no existían se enviara una posición fija para cada extremo.

### Etapa 3: Posibilidad de seleccionar y mover elementos físicos.

En esta etapa se implementa el poder seleccionar y mover elementos físicos cuando la simulación esté detenida.



**Figura 4.** Diagrama de alto nivel para etapa 3.

La figura 4 muestra el diagrama con las clases implementadas.

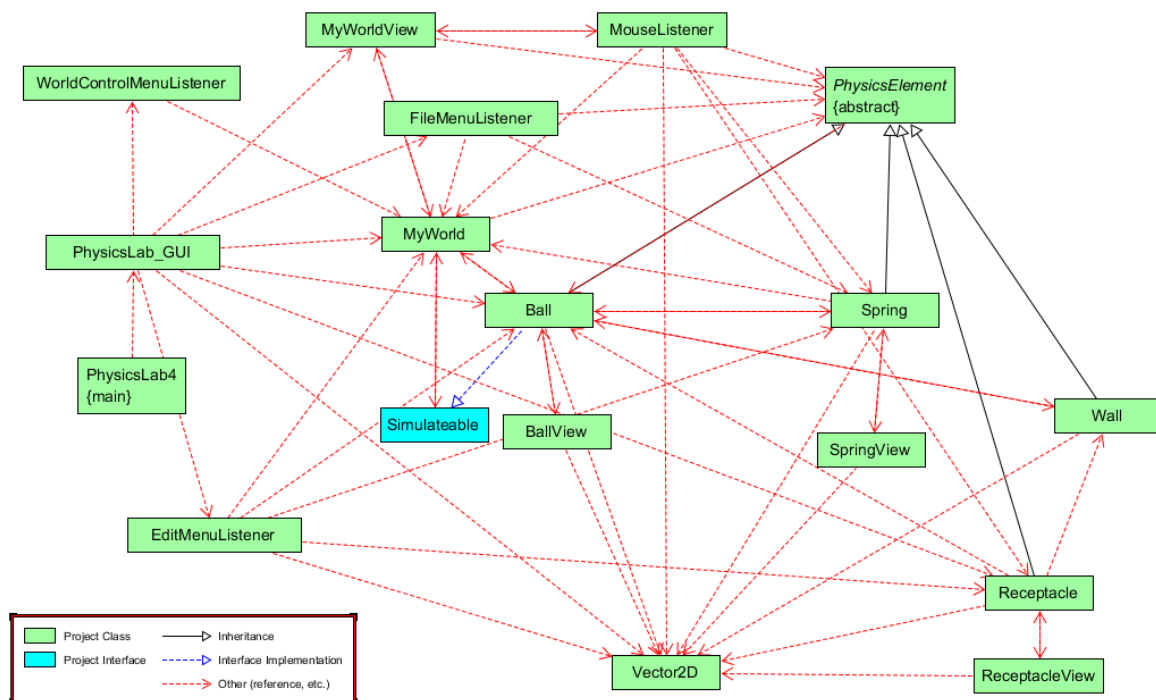
Dificultades y/o observaciones: En esta etapa se tuvieron múltiples dificultades.

- Se tuvo que agregar nuevos atributos al resorte, para las posiciones de sus extremos, ya que ahora estos no estaban necesariamente unidos a bolas.
- Si un extremo del resorte estaba unido a una bola, y luego se acercaba el otro extremo a esa bola, este se unía también. La solución fue agregar una condición antes de asignar la bola al extremo del resorte, de manera que si esa bola ya está unida a este, la bola no se una al otro, también se agregaron atributos booleanos al resorte, que permitían ver que extremo se estaba modificando y así, al soltarse, buscaría unir la bola correspondiente a ese extremo.
- Tanto en el caso del resorte y receptáculo se tenía problemas al usar el método `dragDelta()`, ya que este solo recibía las magnitudes de lo que se había movido, pero no era posible identificar que extremo fue seleccionado. La solución encontrada fue implementar un `dragDelta()` que recibiera la posición del mouse al haberse hecho el click, de este modo, se podía mover el extremos seleccionado.

- En el caso del resorte, al seleccionar un extremo se debía liberar la bola unida a este, así que se debió agregar el método setSelected() que recibiera la posición del mouse, de esta manera, fue posible liberar el extremo adecuado.
- Se agregaron atributos booleanos al resorte, que permitían ver que extremo se estaba modificando y de este modo, al soltarse, buscará unir la bola correspondiente a ese extremo.
- Una dificultad menor, fue el permitir la modificación de los elementos sólo cuando la simulación estuviera detenida. La solución fue sencilla, simplemente se activaba y desactivaba el listener del mouse.

#### **Etapa 4:** Posibilidad de guardar la configuración en un archivo.

En esta etapa se agregó un menú con las opciones ‘Save As...’ y ‘Load’, con las cuales se puede guardar una configuración y cargar una configuración guardada con anterioridad, respectivamente.



**Figura 5.** Diagrama de alto nivel para etapa 4.

La figura 5 muestra el diagrama con las clases implementadas.

Dificultades y/o observaciones: En esta etapa se tuvieron múltiples dificultades.

- La mayor dificultad de esta etapa estuvo en comprender correctamente el uso de las clases JFileChooser, FileInputStream, ObjectInputStream, FileOutputStream y ObjectOutputStream, y la interfaz Serializable.
- Después de mucho estudio y análisis se logró la funcionalidad correcta de la clase FileMenuListener, en la cuál se implementaban las clases dichas en el punto anterior.
- Se usó la clase JFileChooser para abrir una ventana de dialogo y en esta abrir o guardar un archivo, que luego se guardaba en un File con getSelectedFile().
- Luego a un objeto ObjectOutputStream o ObjectInputStream, se le asignaba un FileOutputStream o FileInputStream desde el file obtenido con JFileChooser.
- Luego se escribía o leía en este stream de datos, todos los elementos físicos de la configuración, en un ArrayList de objetos.
- Si se volvía a cargar un archivo, al mover los resortes estos no unían a las bolas. Esto se debía a que al serializar el archivo, el atributo world no se guardaba (ya que ocasionaba problemas, si se intentaba guardar), por lo que al abrir un archivo con resortes, para los resortes no existían bolas en el mundo. La solución implementada fue que, al abrir el archivo y agregar los elementos al mundo, antes de agregar los resortes se le asignara a cada uno la variable del mundo.
- Para que las opciones Load y Save funcionarán estando la simulación detenida, se agregó un atributo booleano a world indicando si la simulación estaba corriendo o no, luego se agregó una condición al seleccionar los ítems del menú.

## **Etapa 5: Se incluye efecto de gravedad y roce del aire.**

En esta etapa se tienen varias bolas en un contenedor, las cuales pueden estar unidas a nada o a uno o más resortes, pero se incluye el efecto de la gravedad y el roce del aire. Se reutiliza casi completamente el código de la etapa anterior.