

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III

Curso 1

Segundo cuatrimestre de 2020

Grupo T10		
Nombre	Padrón	Mail
Bergman, Guido	104030	gbergman@fi.uba.ar
Fábregas, Camilo Ezequiel	103740	cfabregas@fi.uba.ar
Delgado, Nahuel	104078	ndelgado@fi.uba.ar
de Montmollin, Magdalena	99331	mdemontmollin@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Diagramas de clase	2
4. Diagramas de secuencia	3
5. Detalles de implementación	4
6. Excepciones	4

1. Introducción

El presente informe reúne la documentación de la solución del segundo trabajo práctico de la materia Algoritmos y Programación III que consiste en desarrollar una aplicación, de manera grupal, que permite aprender los conceptos básicos de la programación de manera didáctica y visual, en Java utilizando la programación orientada a objetos y las técnicas de TDD e integración continua.

2. Supuestos

Se supuso que el personaje no tiene restricciones espaciales para moverse, es decir, puede moverse en cualquier dirección tantas veces quiera.

Un lápiz puede estar arriba o abajo, no tiene estados intermedios.

3. Diagramas de clase

El programa se separó en dos diagramas de clases para facilitar la lectura de ellos. En las figuras 1 y 2 se pueden observar los diagramas mencionados.

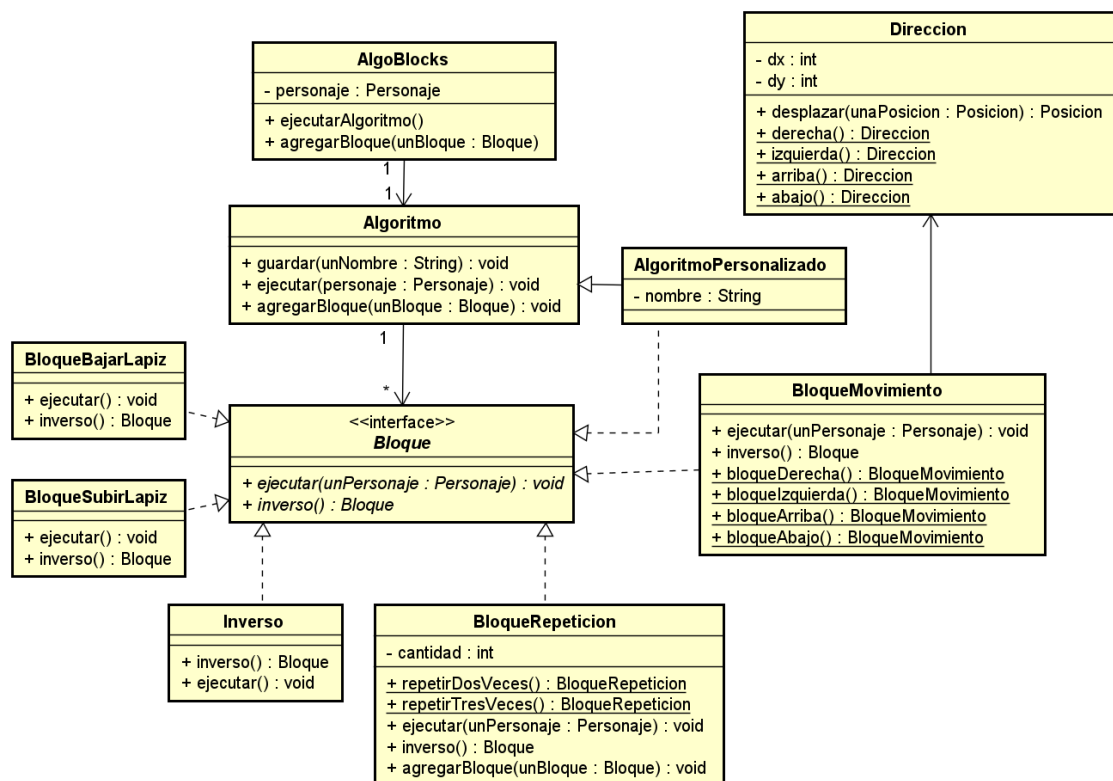


Figura 1: Diagrama de Clases Algoritmo

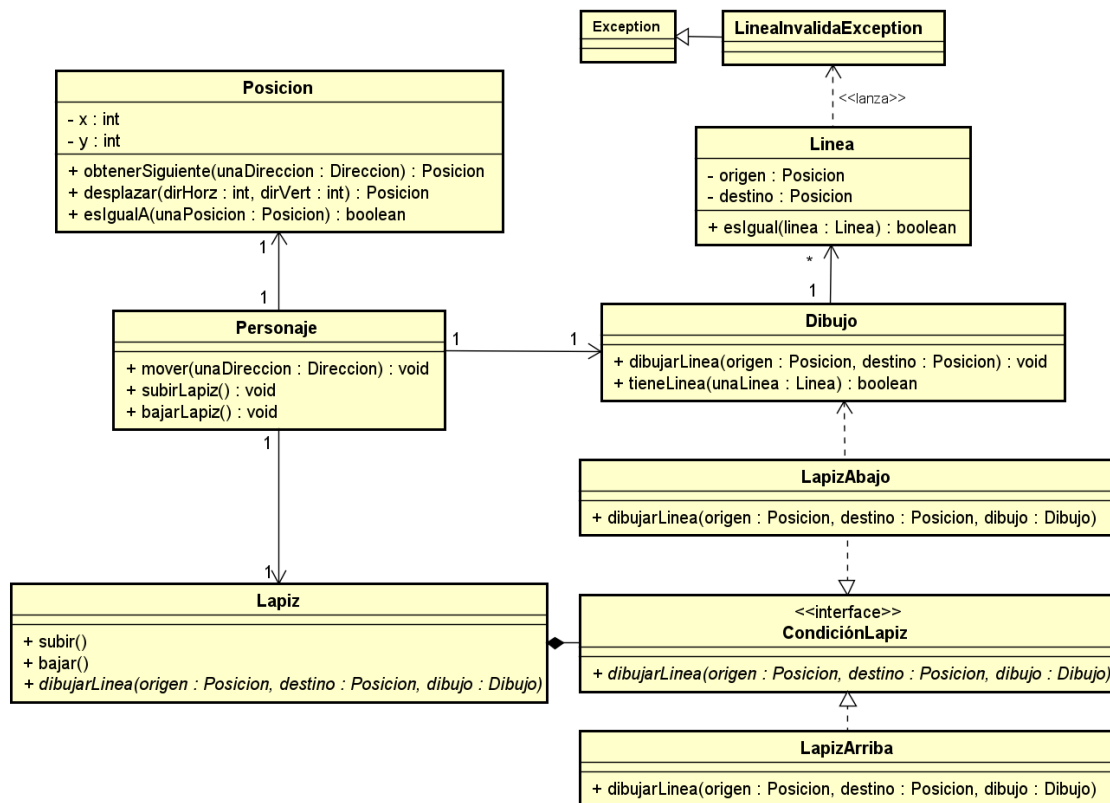


Figura 2: Diagrama de Clases Personaje

4. Diagramas de secuencia

Los diagramas de secuencia representan el caso de ejecución de un algoritmo con un solo bloque de mover a la derecha.

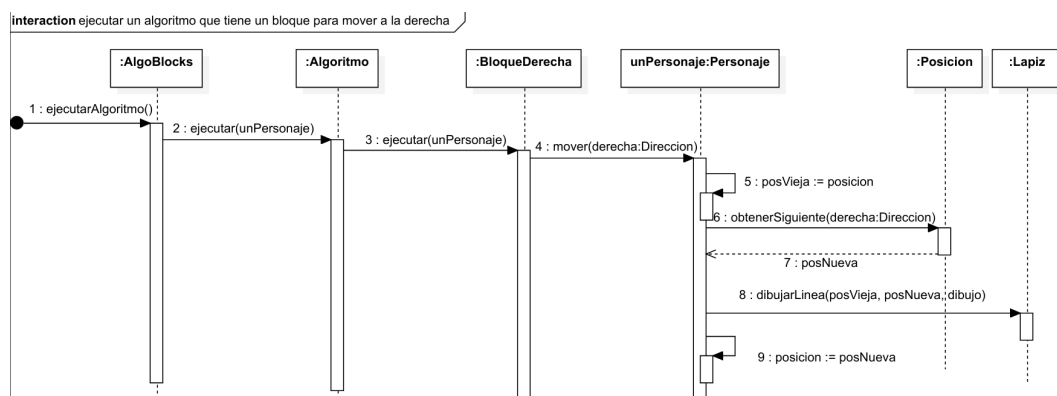


Figura 3: Diagrama de Secuencia de ejecución un algoritmo que tienen un BloqueDerecha

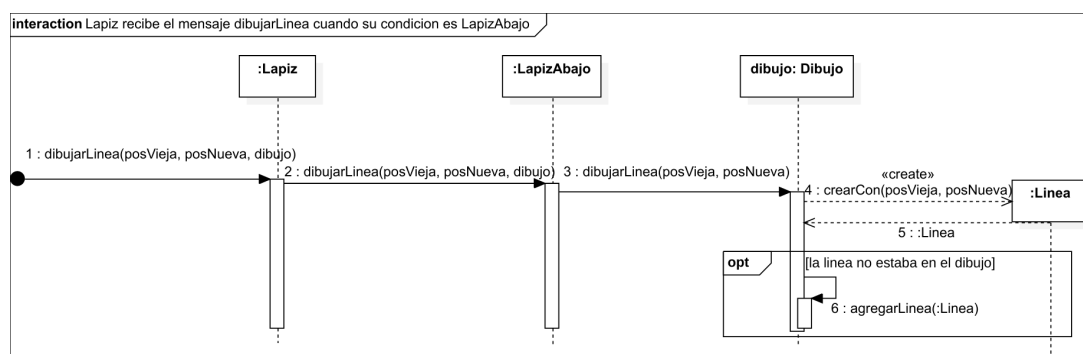


Figura 4: Diagrama de Secuencia de cuando Lapiz recibe el mensaje dibujarLinea con LapizAbajo como condición

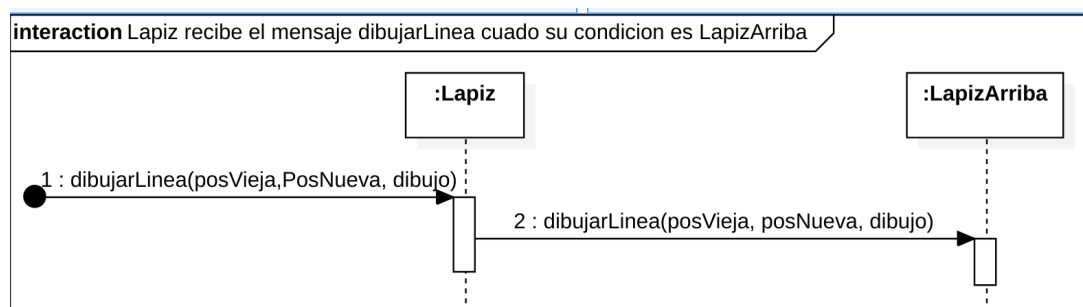


Figura 5: Diagrama de Secuencia de cuando Lapiz recibe el mensaje dibujarLinea con LapizArriba como condición

Como se puede ver, en este caso la linea no se agrega al dibujo.

5. Detalles de implementación

A la hora de implementar el modelo se encontraron se plantearon alternativas sobre como debería comportarse el programa ante ciertas situaciones. En esta sección se mencionan las más relevantes.

No se considera un problema que un **Lapiz** reciba el mensaje **bajar()** si su condición es **LapizAbajo**. Similarmente, no es un problema que reciba **subir()** si su condición es **LapizArriba**. Además, de llamar al método **dibujarLinea(origen: Posicion, destino: Posicion, dibujo: Dibujo)** con el lápiz abajo se crea un objeto **Linea** en la variable **dibujo**, pero un lápiz que esta arriba no crearía una línea visible y por lo tanto no se agrega al dibujo.

Una instancia de **Linea** es igual a otra si sus extremos coinciden, aún cuando sus atributos están cruzados, pues no tienen definido un sentido. Teniendo en cuenta lo anterior, si una **Linea** ya se encuentra en un **Dibujo**, no se la vuelve a agregar a este.

En **AlgoBlocks** tras ejecutar el **Algoritmo**, este mantiene su secuencia de Bloques, no los elimina.

6. Excepciones

LineaInvalidaException Surge al querer crear una línea con la misma posición como origen y fin.