

Trabajo Práctico 2 — Java

[7507/9502] Algoritmos y Programación III
Curso 2
Segundo cuatrimestre de 2020

Alumno 1 :	Araujo, Luis
Número de padrón:	102112
Email:	laraujo@fi.uba.ar
Alumno 2 :	Arroyo, Sebastian
Número de padrón:	100728
Email:	sarroyo@fi.uba.ar
Alumno 3 :	San Martin, Nicolas
Número de padrón:	104320
Email:	nsanmartin@fi.uba.ar
Alumno 4 :	Branko Tintilay Tacacho, Ivan
Número de padrón:	102479
Email:	btintilay@fi.uba.ar

Índice

1. Introducción	2
2. Supuestos	2
3. Modelo de dominio	2
4. Diagramas de Clase	3
5. Diagramas de Secuencia	5
6. Diagrama de Estado	7
7. Diagrama de Paquetes	7
8. Detalles de Implementación	7
8.1. Bloques:	7
8.2. Personaje y su movimiento:	7
8.3. Personaje y el uso del lápiz:	8
8.4. Patrón Strategy, Dibujador, LápizArriba y LápizAbajo:	8
9. Excepciones	8

1. Introducción

En el presente informe se documentará la implementación del trabajo práctico 2 de la materia Algoritmos y Programación III, el cual consiste en desarrollar un juego orientado al aprendizaje de los conceptos básicos de la programación, mediante el armado de algoritmos usando una serie de bloques con distintas funcionalidades que se verán reflejadas a través de un dibujo en el tablero. El lenguaje de programación que se utilizó es Java, siguiendo el paradigma de la Programación Orientada a Objetos (POO).

2. Supuestos

En esta sección se explicitarán aquellos factores que obviaremos en el desarrollo del proyecto.

1. El personaje no podrá salirse de los límites del tablero y que este no quede a la vista del usuario.
2. Al crearse un personaje este iniciará, por defecto, con el lápiz arriba.

3. Modelo de dominio

En cuanto al diseño implementado, se consideró como clase principal a Tablero, que es una interface de la que hacen uso las clases (que llamaremos sectores) SectorBloques, SectorDibujo y SectorAlgoritmo.

El primero se encargará de contener y mostrar los bloques con los que puede interactuar el usuario, que al usarlos se enviarán a SectorAlgoritmo, quien almacenará la información de los bloques elegidos. El personaje será quien utilice los bloques de este último sector para plasmarlos en el sector de dibujo.

4. Diagramas de Clase

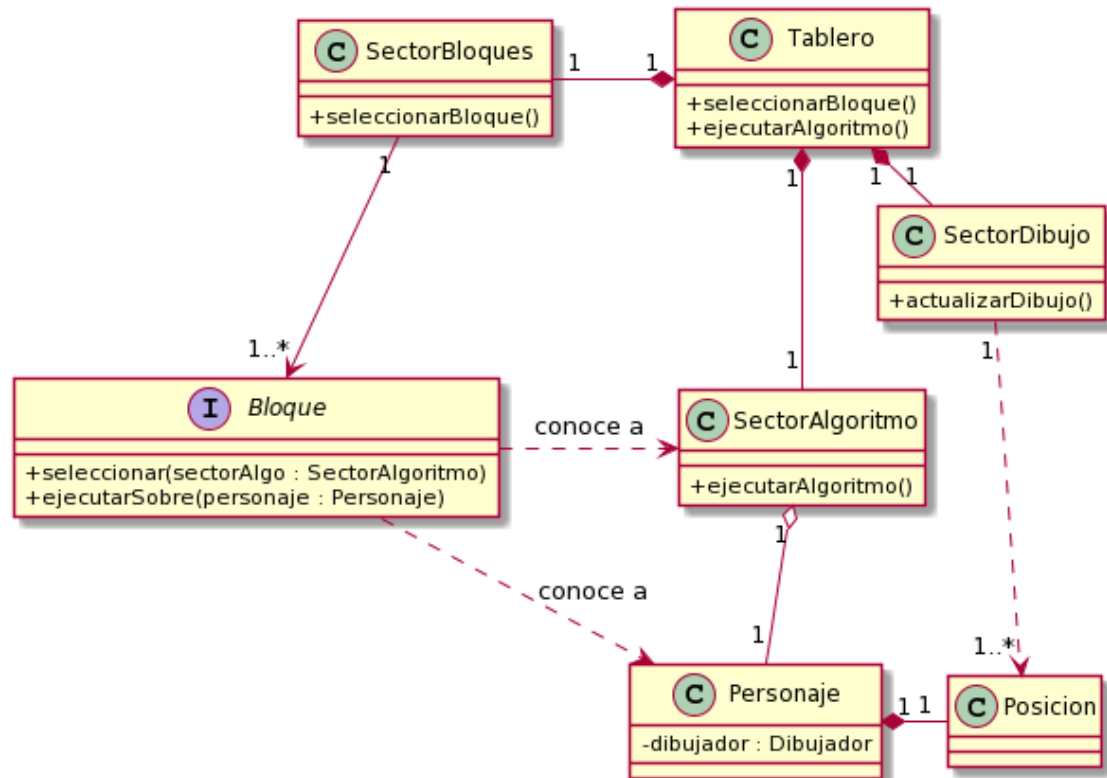


Figura 1: Diagrama principal del modelo a seguir, donde se muestra la Relación del Tablero con los Sectores Bloques, Algoritmo y Dibujo junto con el Personaje.

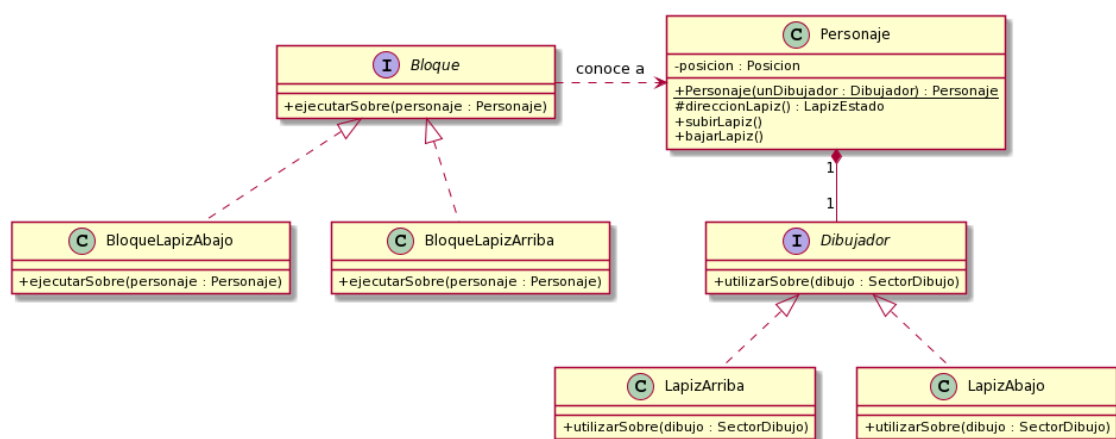


Figura 2: Diagrama sobre la relación de Personaje con la interfaz Dibujador y las clases LapizArriba y LapizAbajo que la implementan, también se representan los bloques que suben y bajan el lápiz.

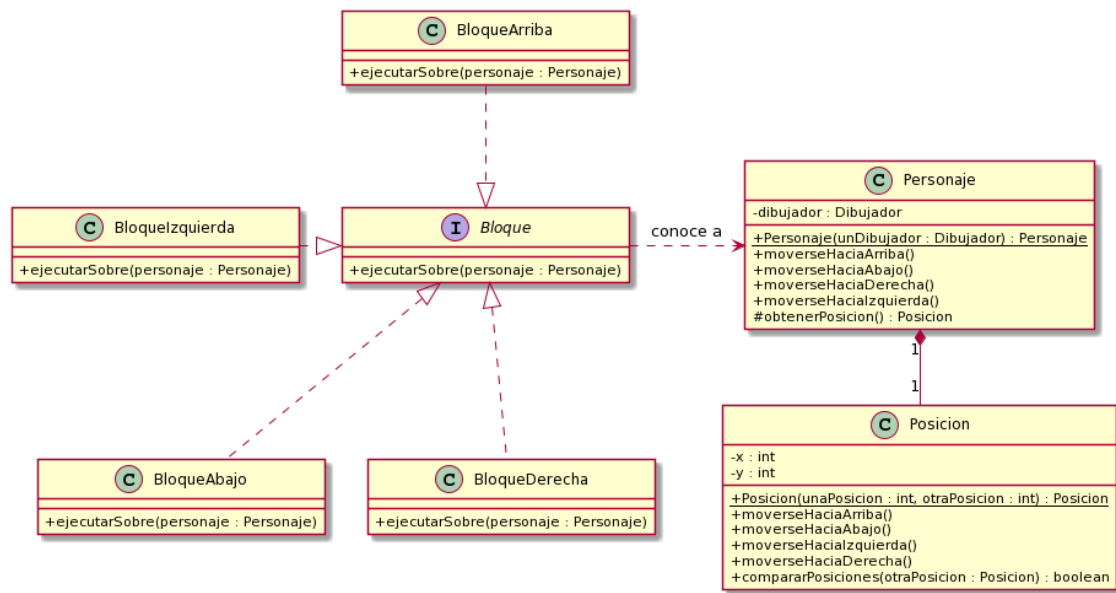


Figura 3: Diagrama sobre la relación de Personaje con la clase Posición, también se representan los bloques que se encargan del movimiento del Personaje.

5. Diagramas de Secuencia

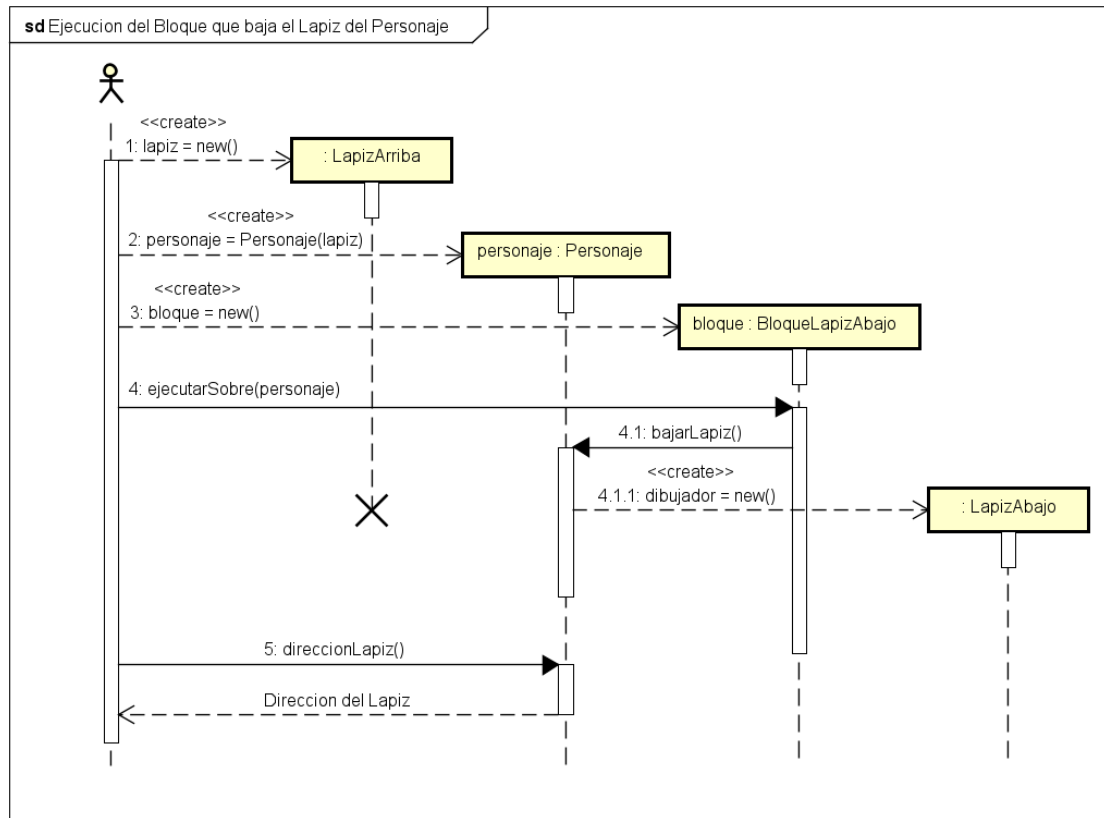


Figura 4: Diagrama sobre la ejecución de un algoritmo con un bloque de bajar el lápiz.

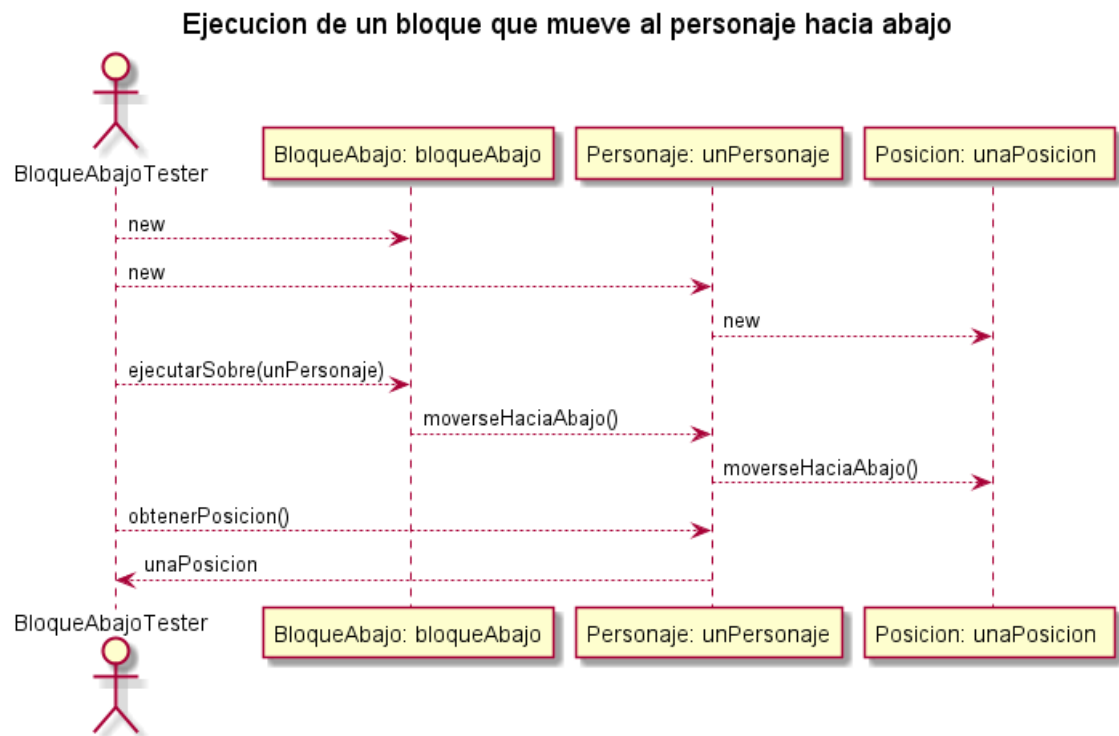


Figura 5: Diagrama sobre el movimiento del Personaje al ejecutarse un bloque de mover hacia abajo.

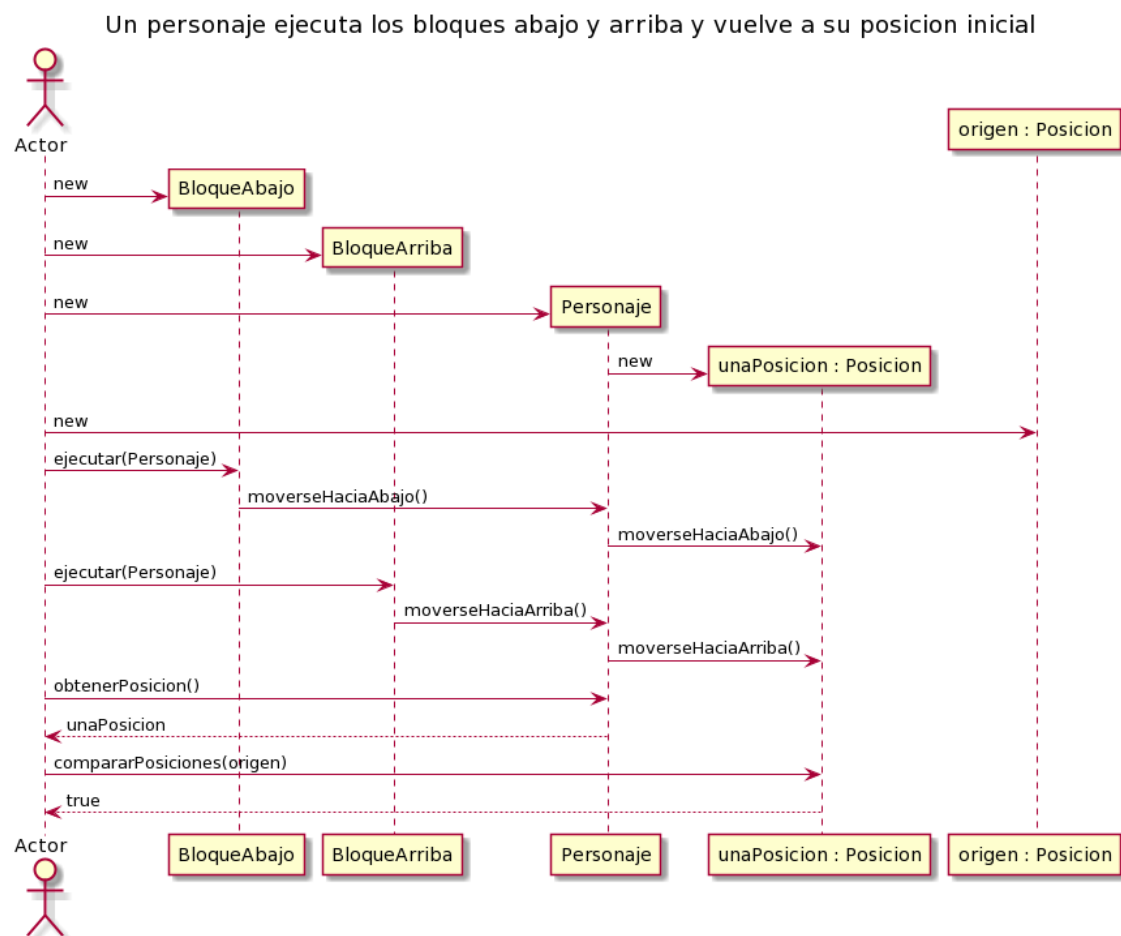


Figura 6: Diagrama sobre el movimiento del Personaje al ejecutarse un bloque de mover hacia abajo y mover hacia arriba con los cuales vuelve a su posición inicial.

6. Diagrama de Estado

7. Diagrama de Paquetes

8. Detalles de Implementación

8.1. Bloques:

Para el uso de Bloques que suben, bajan y que mueven al Personaje, decidimos implementar una interface con este nombre, de la cual heredan los cuales ejecutan su funcionalidad relacionándose con el Personaje mediante un mensaje.

8.2. Personaje y su movimiento:

El personaje instanciará internamente, al momento de su creación, un objeto de la clase *Posición*, la cuál determina la posición en la que este se encuentra. La misma, será la encargada de realizar los movimientos del personaje dentro del tablero, movimiento que son indicados por los bloques BloqueArriba, BloqueAbajo, BloqueDerecha y BloqueIzquierda.

8.3. Personaje y el uso del lápiz:

El personaje recibirá como parámetro del constructor de clase inicialmente una instancia del objeto de la clase `LapizArriba`. El personaje tiene la capacidad de subir y bajar su lápiz mediante métodos, creando en el camino instancias correspondientes a las clases `LapizAbajo` y `LapizArriba`. Las mismas tienen distinto comportamiento relacionándose con el `SectorDibujo`. El uso del lápiz, es decir la capacidad de dibujar, estaría sujeta a la capacidad de movimiento del Personaje.

8.4. Patrón Strategy, Dibujador, LapizArriba y LapizAbajo:

Implementamos la interface *Dibujador*, de la cual heredaran las clases `LapizAbajo` y `LapizArriba`. Esta misma sirve como una herramienta de dibujo y para eso decidimos implementarla, con un método `utilizarSobre`, con el cual `LapizAbajo` y `LapizArriba` interactuaran con el `SectorDibujo` en formas diferentes. Para realizar esto nos basamos en utilizar el patrón Strategy, el cual busca tener un comportamiento que sea solucionado en tiempo de ejecución, ya que a lo largo de la vida del personaje este deberá poder subir o bajar su lápiz. Estas últimas capacidades estarían indicadas por los bloques de `BloqueSubirLapiz` y `BloqueBajarLapiz`.

9. Excepciones