**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**FACULTAD DE INGENIERIA**

**ESCUELA DE CIENCIAS Y SISTEMAS**

**TORRES DE HANOI**

**MANUAL TECNICO**

**ELABORADO POR:**

JOSUÉ DANIEL ROJCHE GARCIA

**FECHA:**

07/10/2021

**INDICE**

Objetivos……………………………………………………………………………………2

Especificación Técnica……………………………………………………………………3

Lógica del programa……………………………………………………………………….5

**OBJETIVOS**

**GENERAL**

Entender la codificación del programa y los requerimientos necesarios para su realización.

**ESPECIFICOS**

* Realizar un software compatible con varios sistemas operativos.

**ESPECIFICACION TECNICA**

**REQUISITOS DE HARDWARE**

El sistema puede ser utilizado en computadoras con los siguientes requerimientos mínimos.

* Memoria RAM : 2.00 GB
* Procesador: 1 giga Hertz (GHz) o más rápido con 2 o más núcleos en un procesador de 32 o 64 bits.

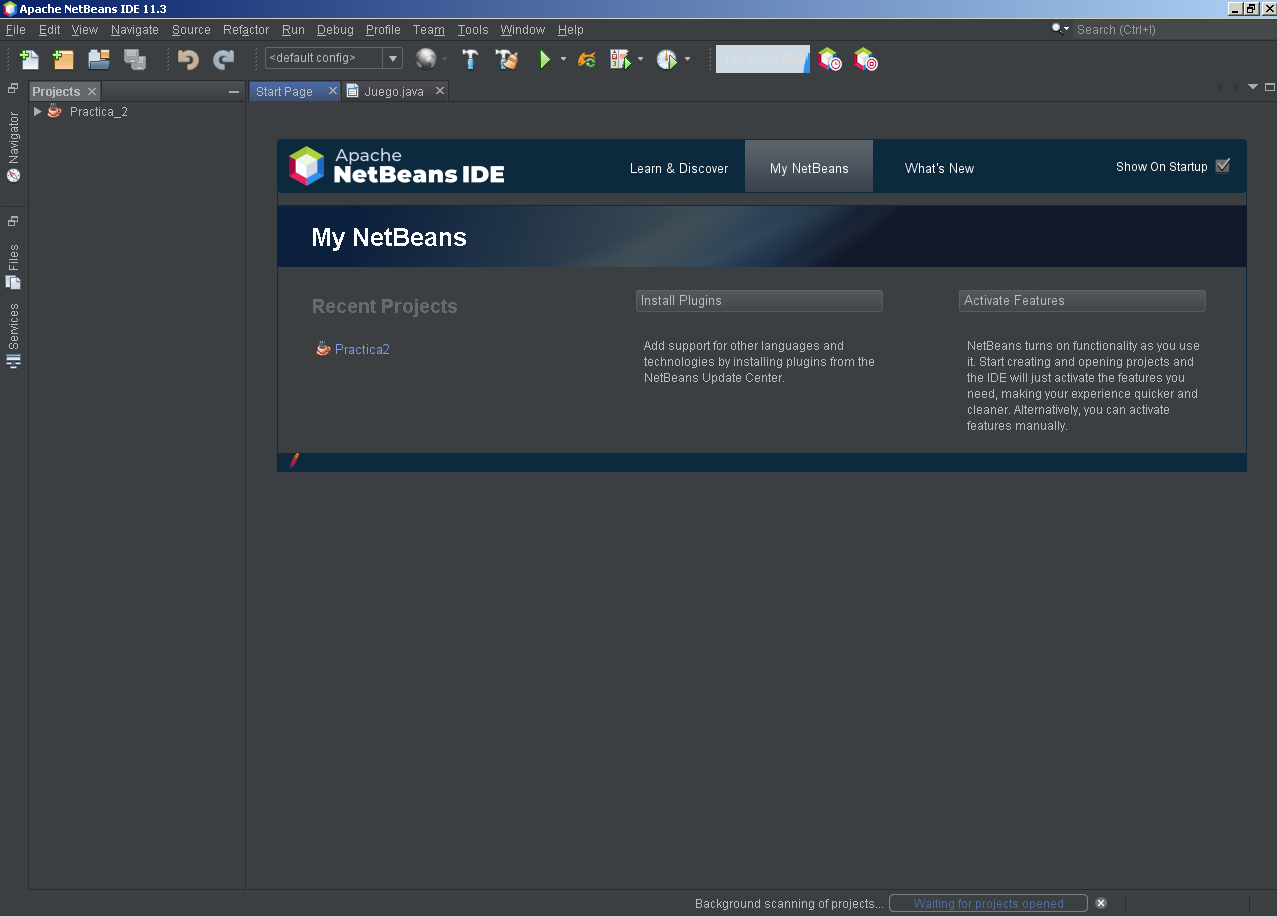
**REQUISITOS DE SOFTWARE**

El sistema puede funcionar con sistemas operativos de Windows 7 en adelante, y con sistemas operativos de las distribuciones Linux los cuales pueden ser de 32 bits o 64 bits.

El lenguaje de programación y el IDE utilizado para la realización del software son los siguientes:

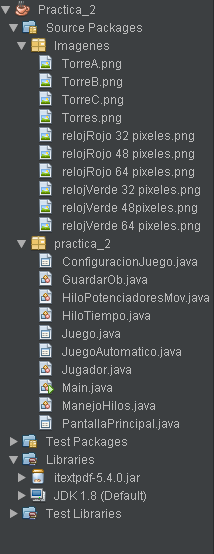
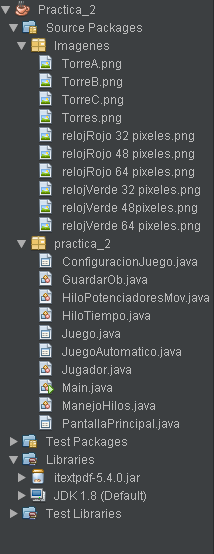
* JAVA 8 Update 111
* JAVA SE Development Kit 8 Update 111
* Apache NetBeans IDE 11.3





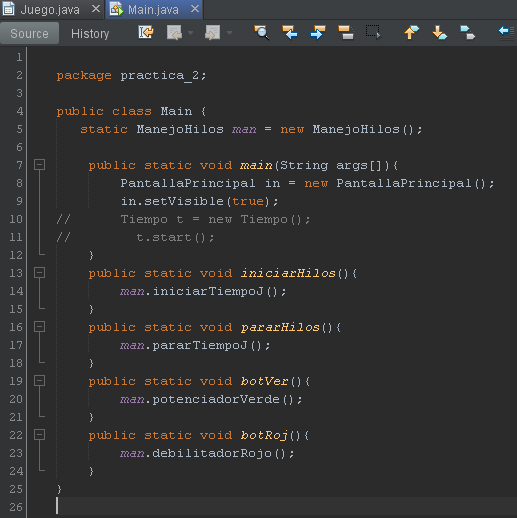
**LOGICA DEL PROGRAMA**

En el proyecto se trabajó con 2 paquetes uno para la realización de las clases a utilizar y el otro para el manejo de las imágenes utilizadas para el software.

****

En el paquete de las clases se tiene la clase principal correspondiente al Main el cual inicia la ejecución del programa.

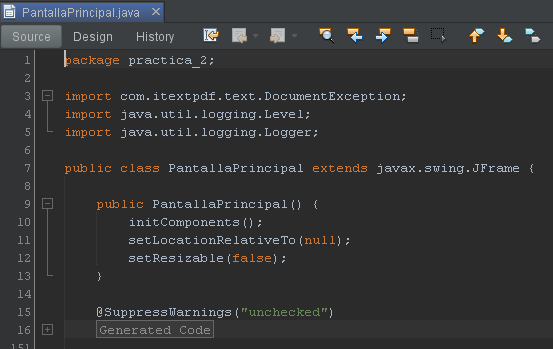
En la clase Main se realiza la instancia de la clase ManejoHilos y de la pantalla principal.



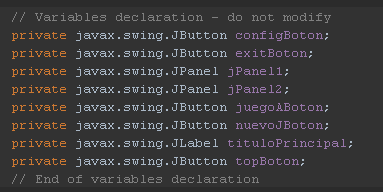
En la clase de pantalla principal se realizó con Drag and Drop agregando los botones y labels.



Se tiene el método constructor donde se inicializa los componentes de dicha clase.

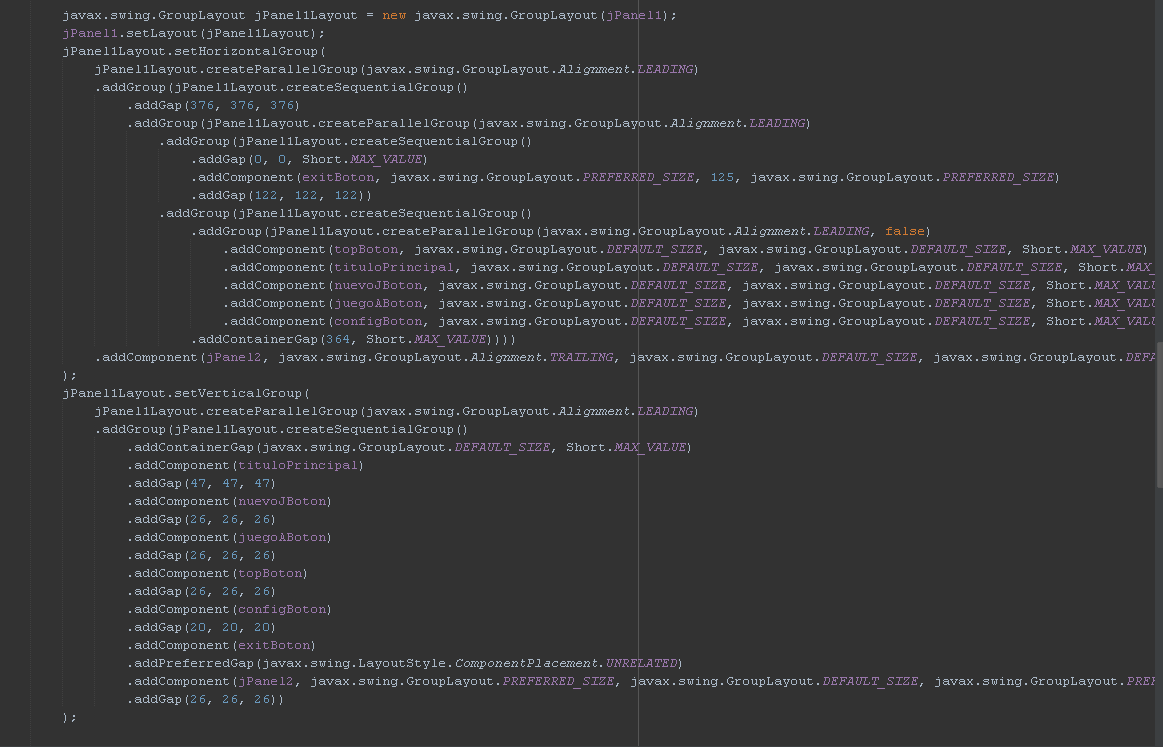


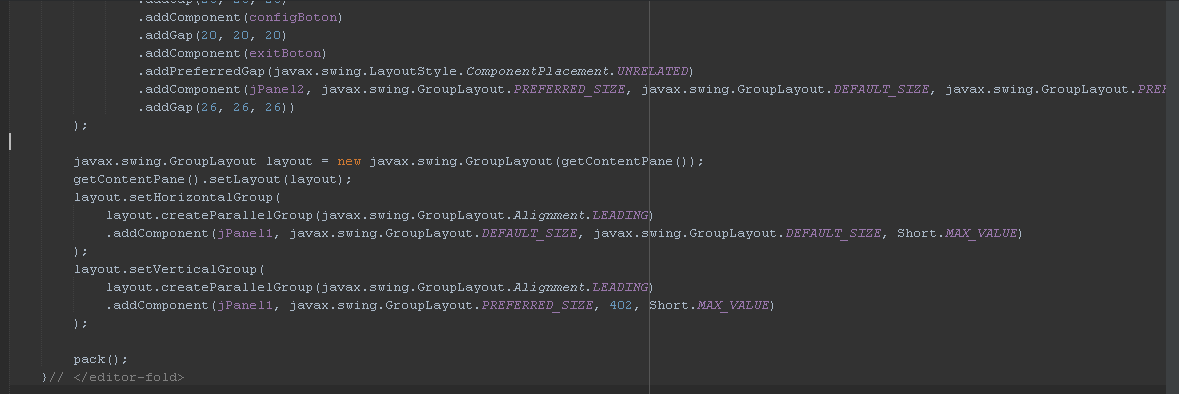
La declaración de las variables y la inicialización de los componentes los realiza automáticamente el IDE, por lo tanto no es posible editar directamente el código sino que se debe dar dichas propiedades desde Drag and Drop.





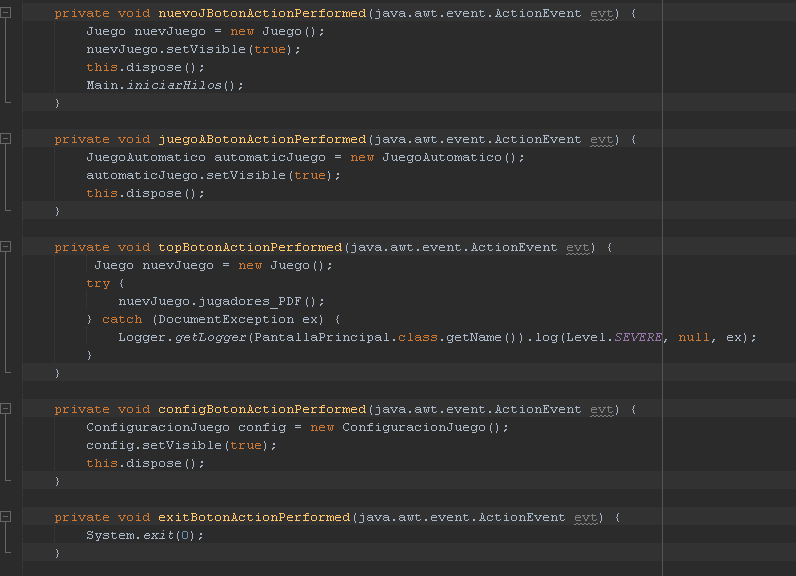




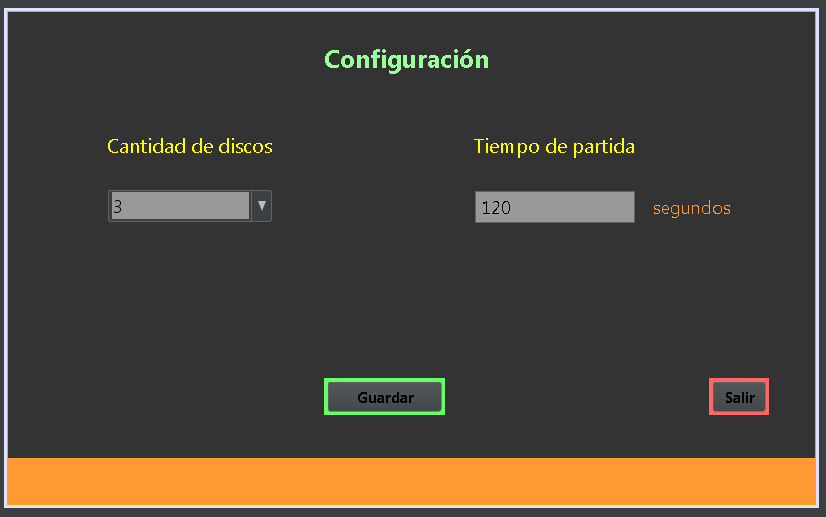


Los métodos para las acciones de los botones contienen las instancias de las clases que corresponden a lo que se requiere en cada uno.

El botón de correspondiente al juego nuevo hace la instancia de la clase Juego y al abrir dicha clase se oculta la clase de la pantalla principal y inicia la ejecución del hilo que maneja el tiempo que durará el juego.

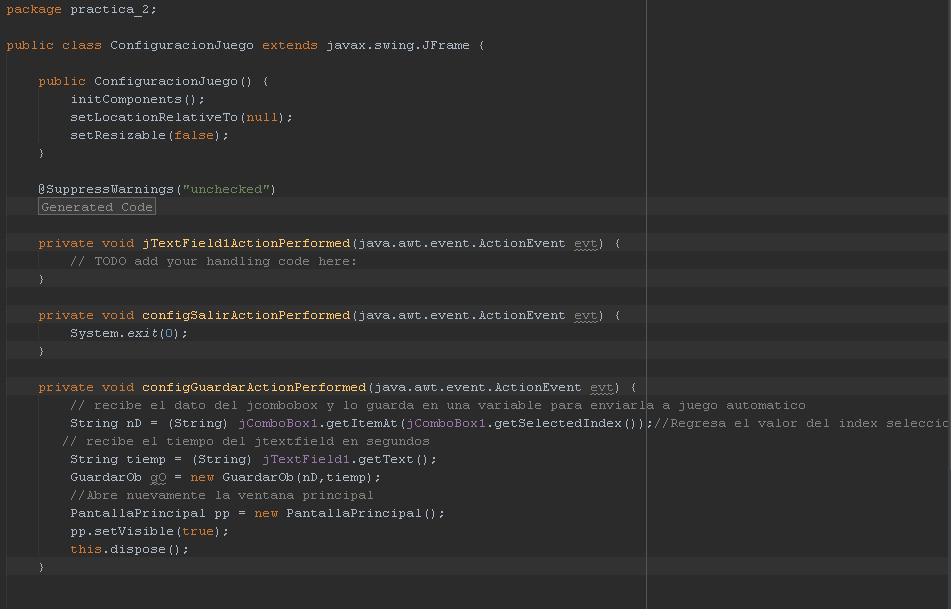


La clase de configuración maneja la cantidad de discos que se utilizará en el juego y también los segundos que se requiera.

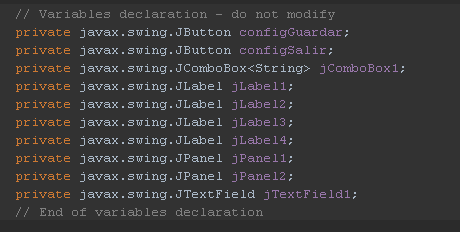


El método constructor inicializa el método de los componentes.

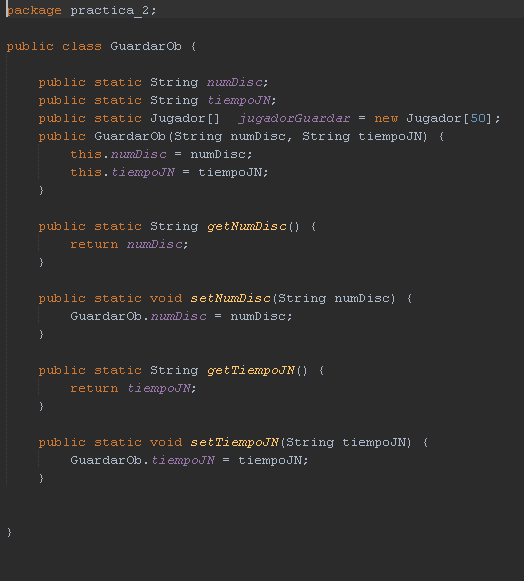
El botón de guardar los datos envía los datos obtenidos de los componentes a la clase de guardar objetos.



El IDE inicializa automáticamente las variables de los componentes utilizados.



La clase de guardar objetos almacena las variables del número de discos a utilizar en el juego, el tiempo y el arreglo de jugadores que almacena el nombre del jugador el tiempo restante y la cantidad de movimientos que realizó el jugador para ganar la partida

****

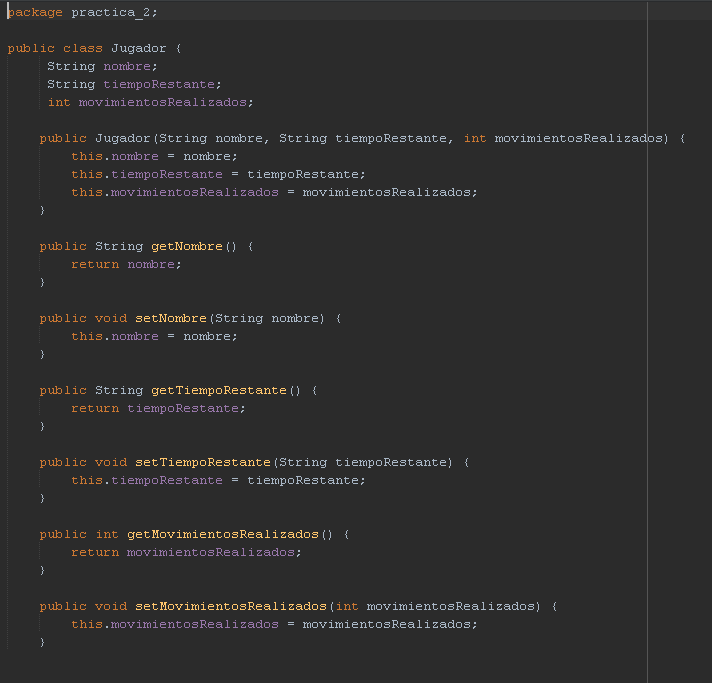
La clase HiloTiempo maneja el tiempo en cuenta regresiva en segundos y envía el tiempo restante al objeto jugador para la respectiva realización del top 5.

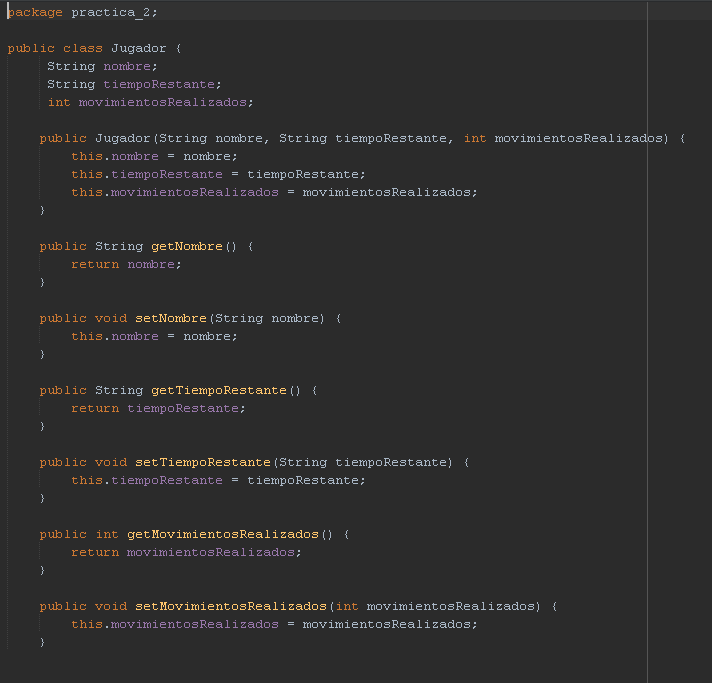


La clase del manejo de hilos inicia el flujo del hilo, lo detiene, y envía la potencia del botón para dar los 10 segundos al juego, así como el debilitador que quita los 10 segundos.

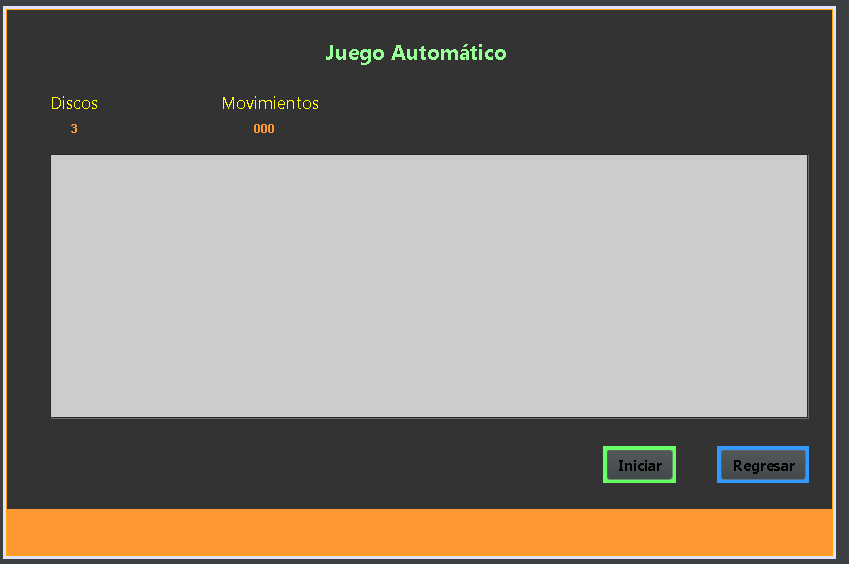


La clase jugador maneja los datos del jugador que sirve para realizar el top5, en el cual se obtienen los datos a través del método constructor y sus respectivos métodos getters y setters que sirven para recibir y enviar datos.

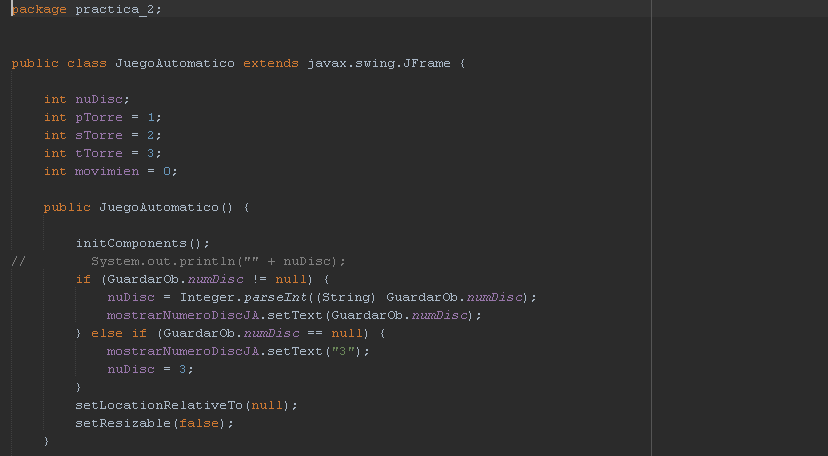




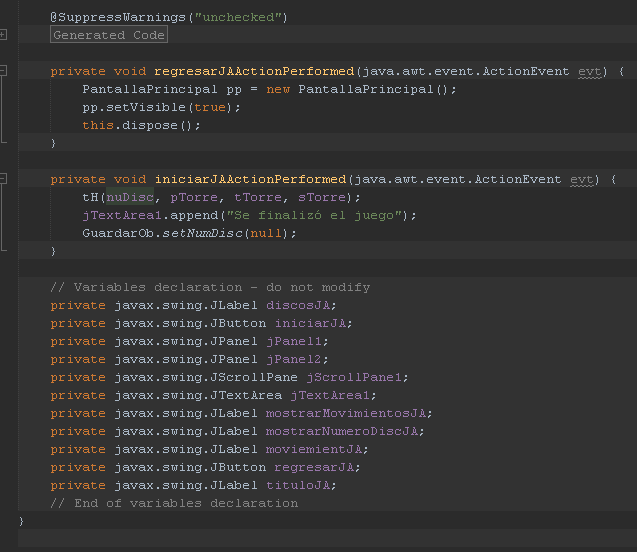
La clase juego automatico muestra la cantidad minima de movimientos para realizar un juego perfecto, asi como también muestra el algoritmo del juego en el espacio.



Esta clase utiliza métodos recursivos para la realización del juego automático utilizando las variables para cada una de las torres y la cantidad de discos y los movimientos realizados durante la ejecución de la misma.



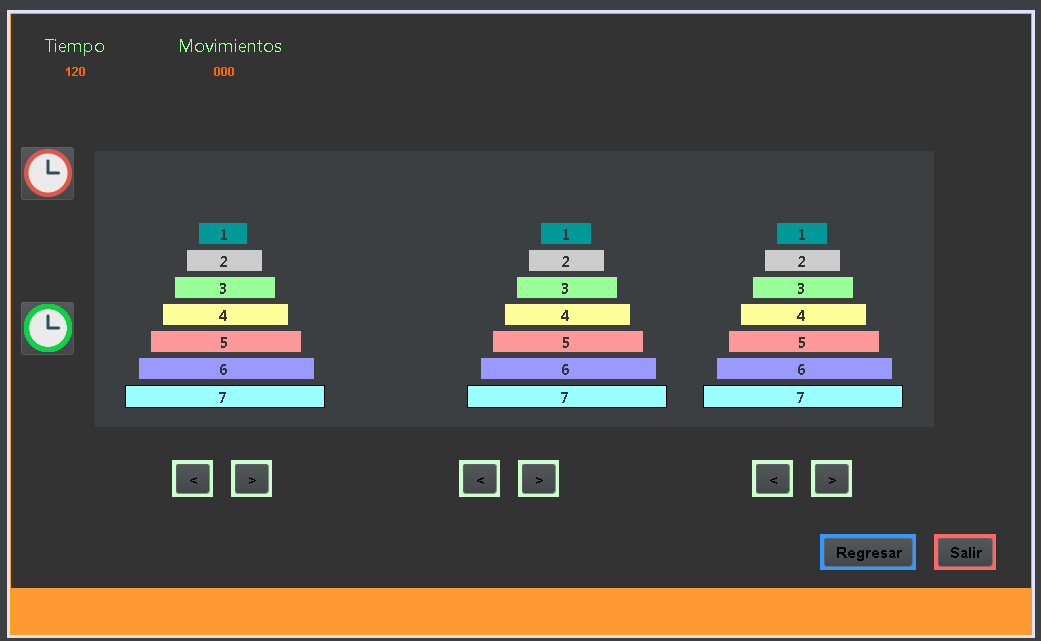




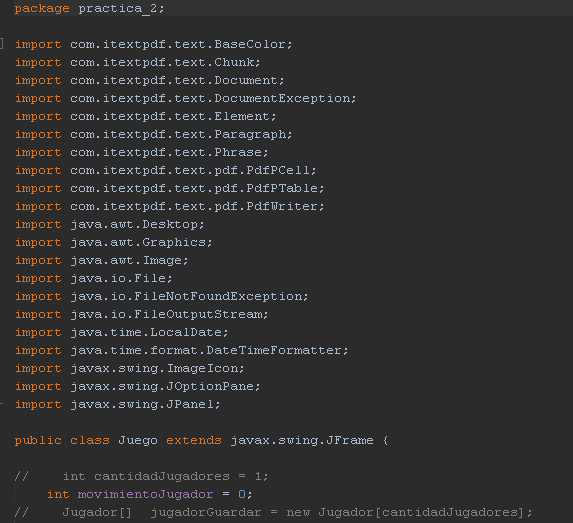
La clase juego es la que más métodos contiene ya que debe realizar más acciones que las otras clases.

En este caso se realizó los botones para las direcciones del movimiento de los discos, botón para salir y regresar, el botón para el potenciador y el botón para el debilitador.

Para los discos se utilizó Jlabels y se realizaron 21 debido a que son 7 discos como máximo y porque son 3 torres.



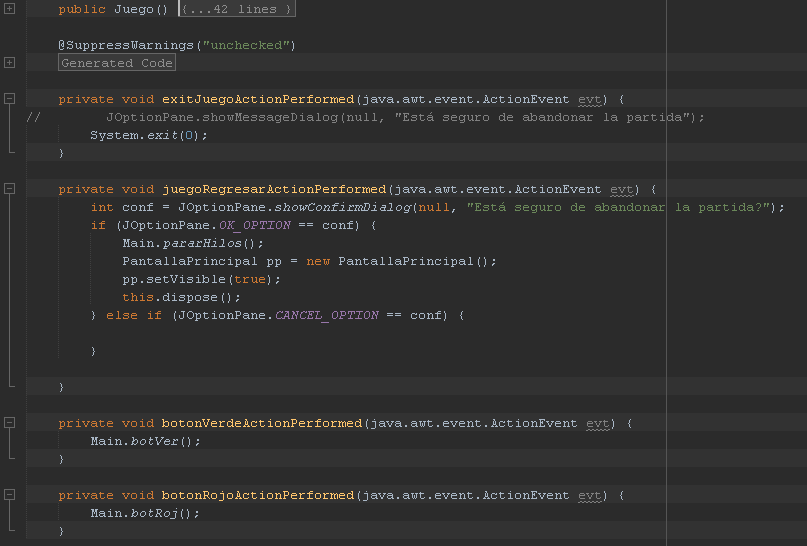
De igual manera que en las otras clases la clase de juego inicializa las variables automáticamente y los componentes del mismo pero en esta clase se agrega la librería necesaria para mostrar el top5 de los jugadores que menor tiempo hicieron para ganar el juego en formato pdf.



En el método constructor se agregan varias validaciones para que el juego funcione si no se realiza alguna configuración previa y también para realizar la visibilidad de los discos a utilizar de acuerdo a la configuración por defecto y la configuración personalizada.

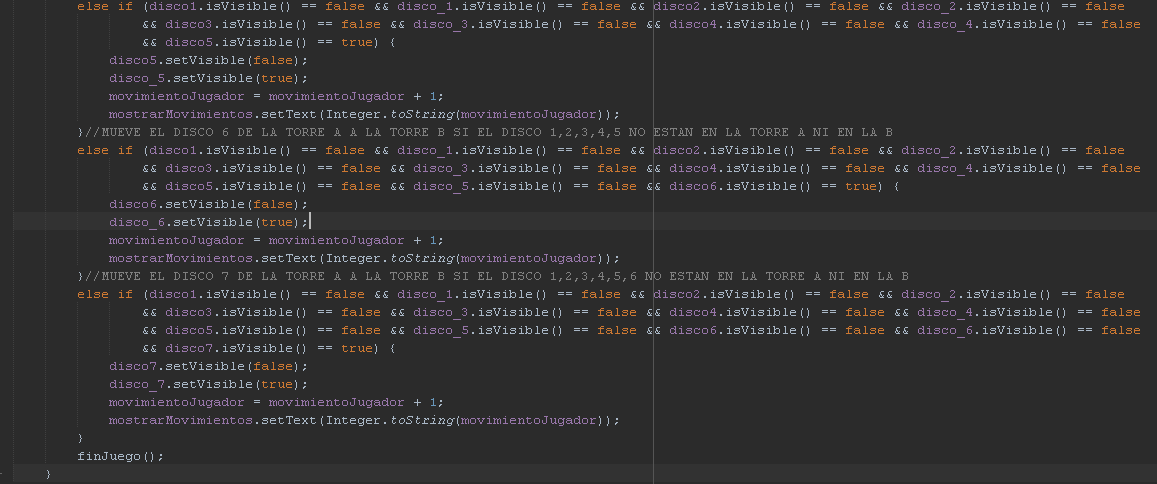


Los métodos para las acciones de los botones de salir, regresar y de los botones para el potenciador y el debilitador.

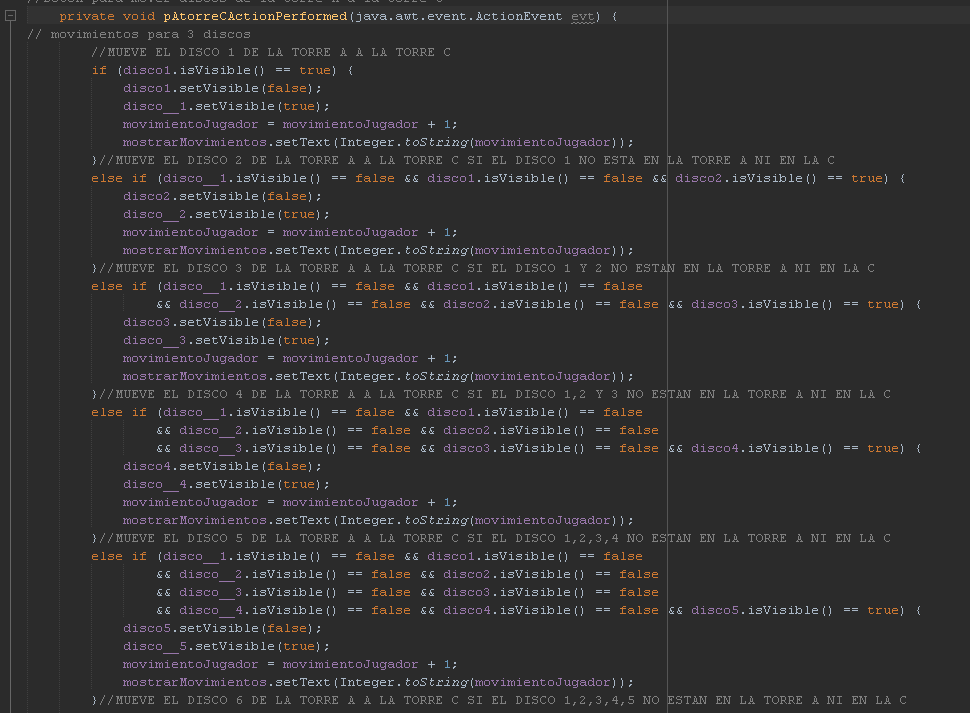


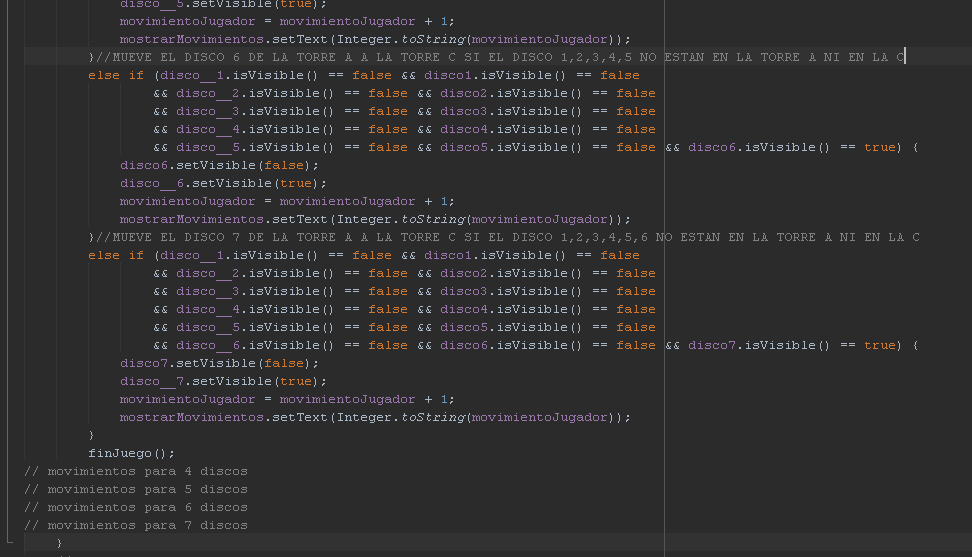
El método para la acción del botón para mover los discos de la primer torre a la segunda, utiliza validación de la visibilidad de los JLabels.



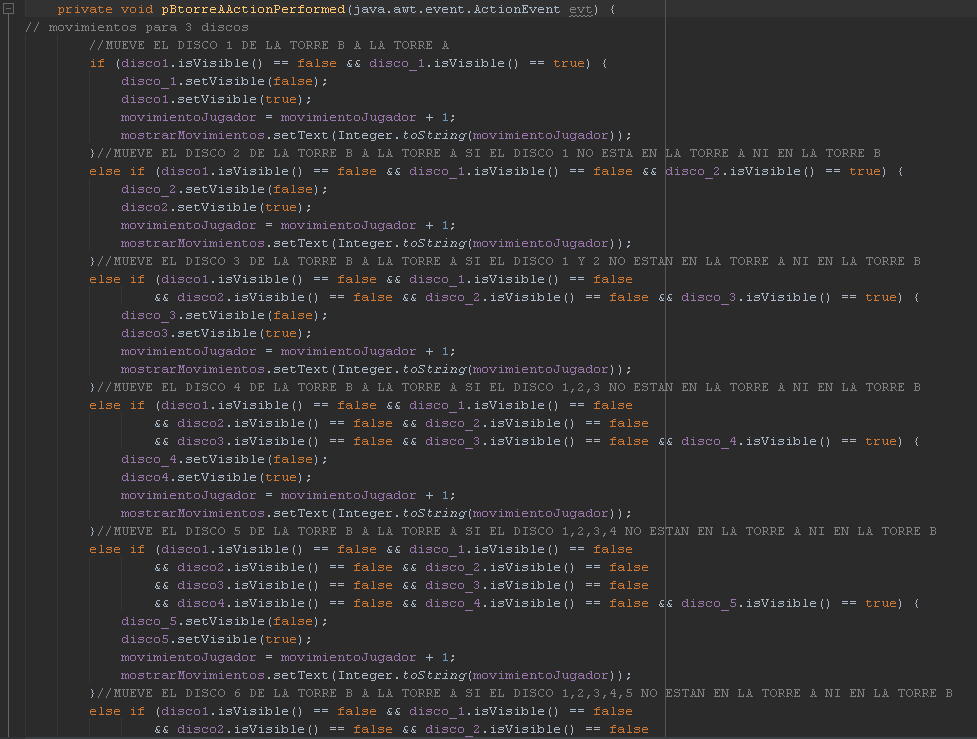


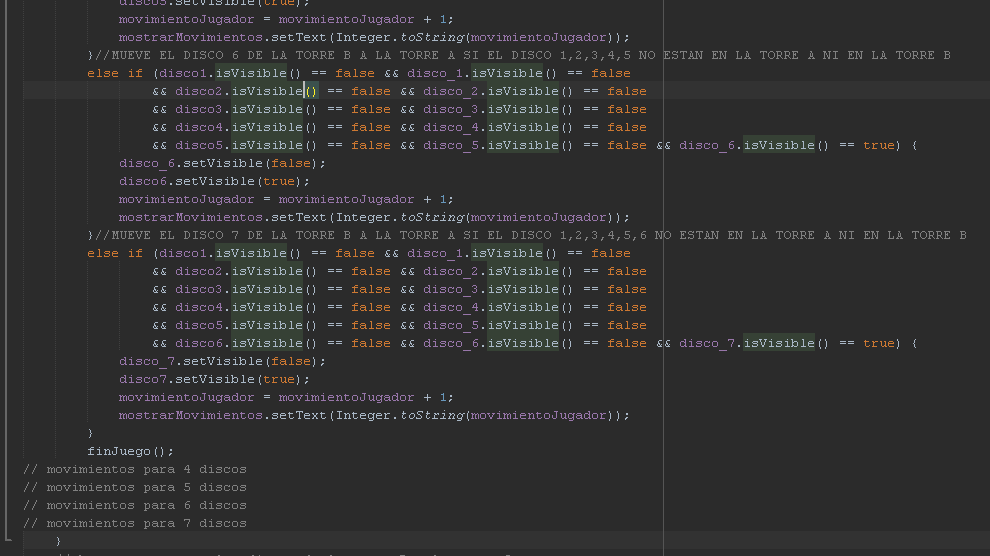
El método para la acción del botón para mover los discos de la primer torre a la tercera, utiliza validación de la visibilidad de los JLabels.





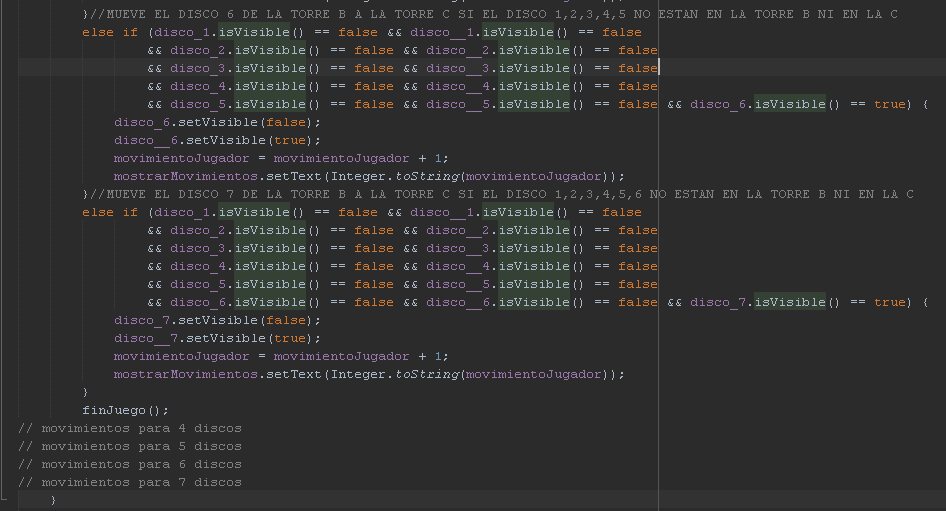
El método para la acción del botón para mover los discos de la segunda torre a la primera, utiliza validación de la visibilidad de los JLabels.



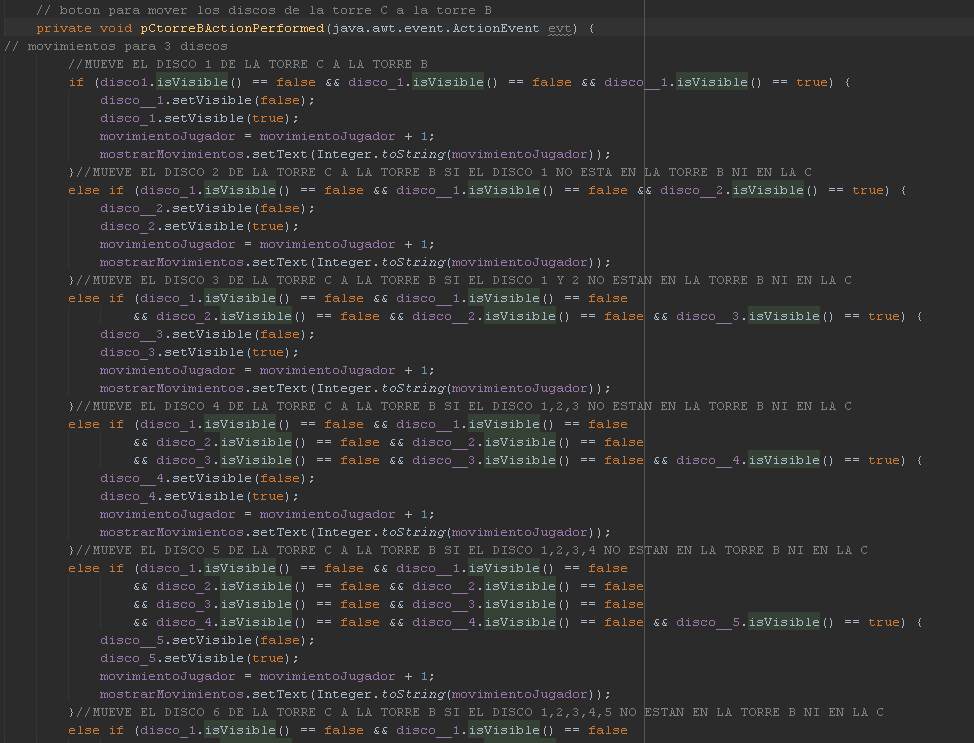


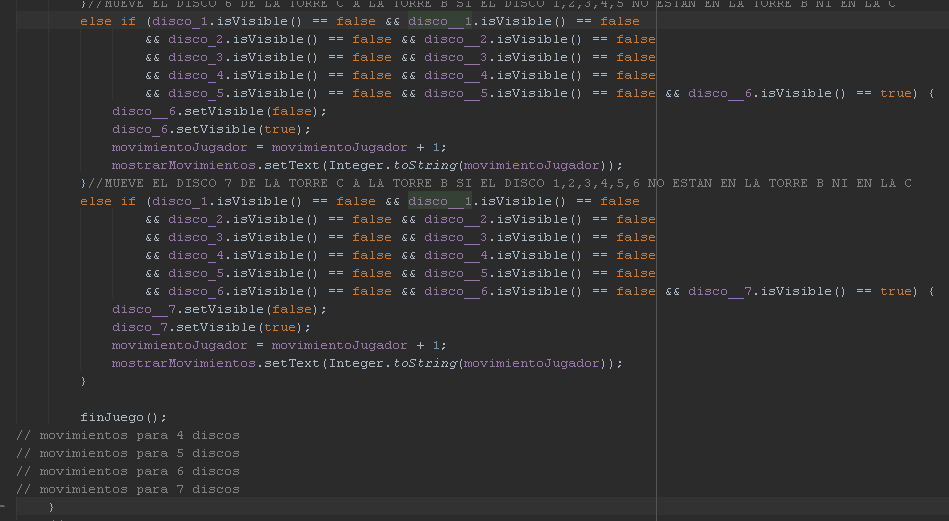
El método para la acción del botón para mover los discos de la segunda torre a la tercera, utiliza validación de la visibilidad de los JLabels.





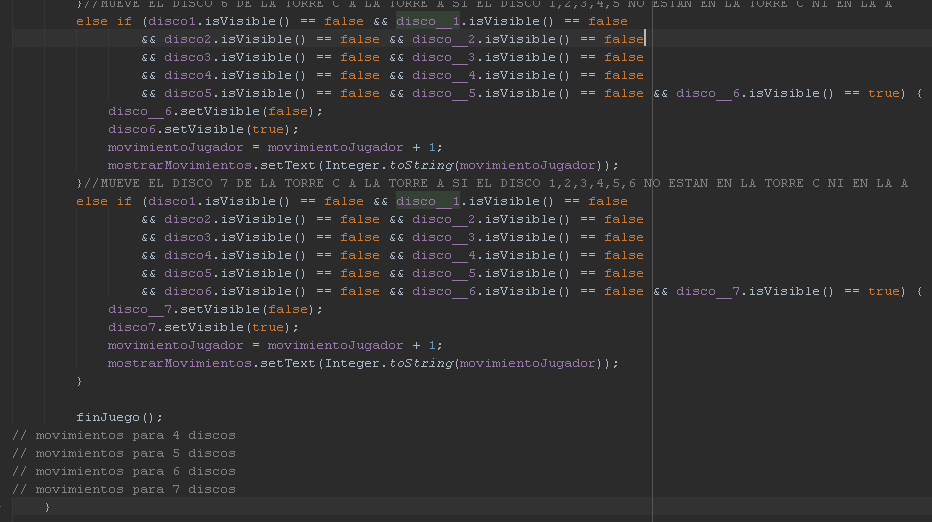
El método para la acción del botón para mover los discos de la tercera torre a la segunda, utiliza validación de la visibilidad de los JLabels.





El método para la acción del botón para mover los discos de la tercera torre a la primera, utiliza validación de la visibilidad de los JLabels.

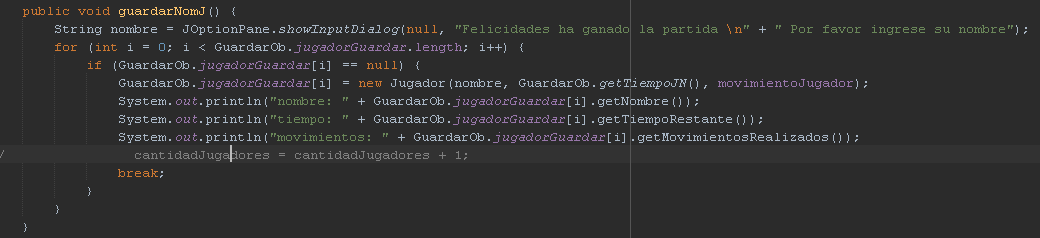




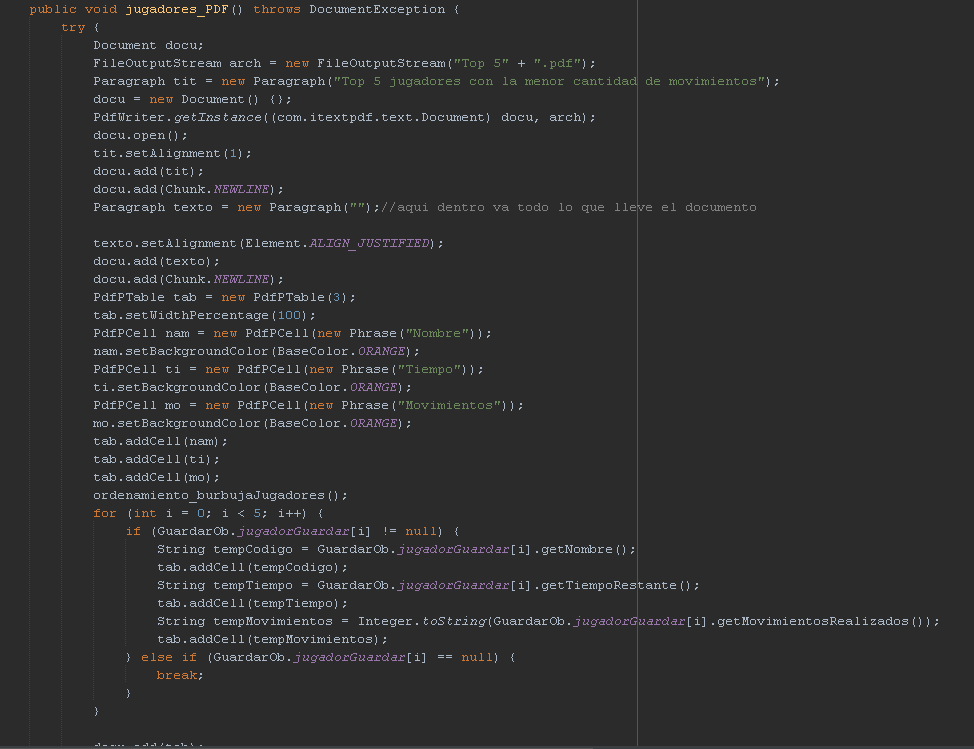
Método para cuando todos los discos se encuentran en la última torre y finaliza el juego.

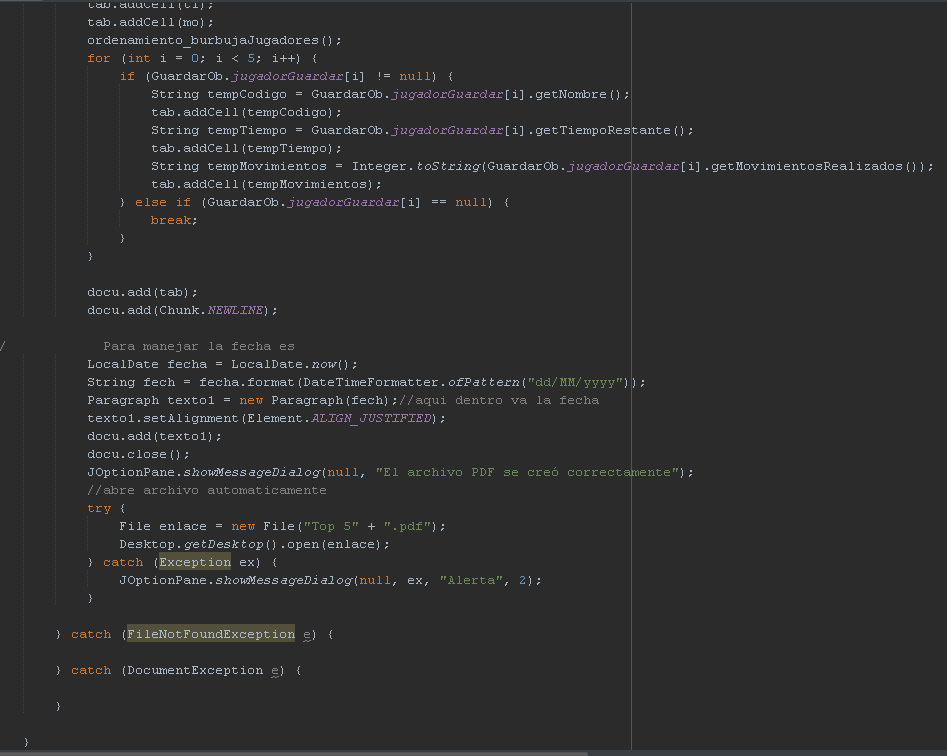


Método para guardar los datos del jugador.



Método para la realización del documento con el top5 de los jugadores que ganaron el juego con menos movimientos.





Método que realiza el ordenamiento burbuja para la realización del top5.

