LABORATORIO 3 POOB Nicolas Aguilera y Daniel Walteros

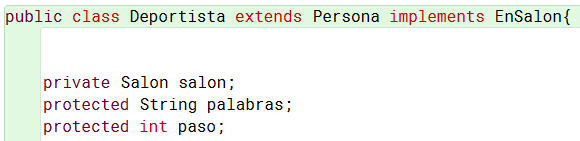
**Conociendo**

1. Hay dos paquetes (aplicación y presentación), cinco clases (salón, Persona, Deportista, Bola, BodyTicGUI)
2. FALTA

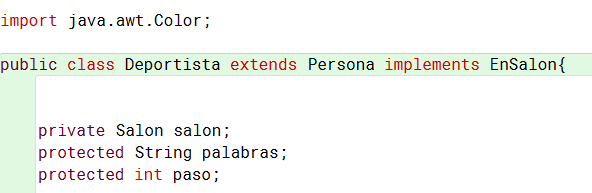
**Arquitectura General**

Paquete es un conjunto de clases e interfaces relacionadas entre sí. Para guardar algo en el paquete basta con poner la instrucción Package seguida del nombre del paquete.





Import es un comando que permite importar clases de un paquete específico, para usarlo se denomina la ruta de la clase dentro del paquete.

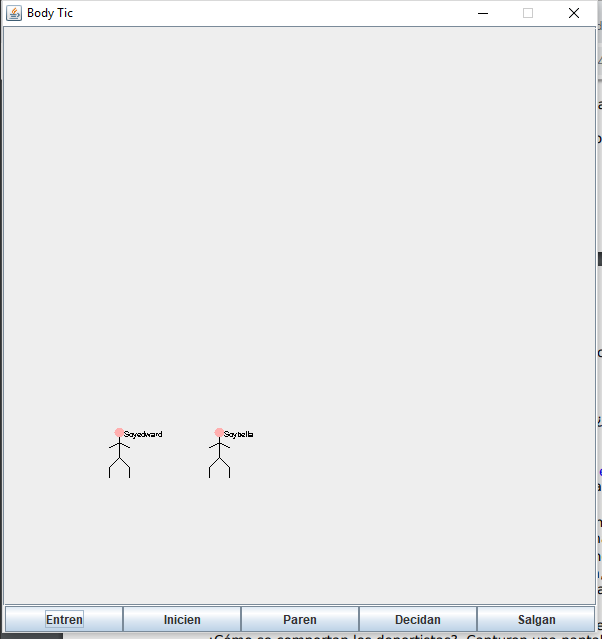
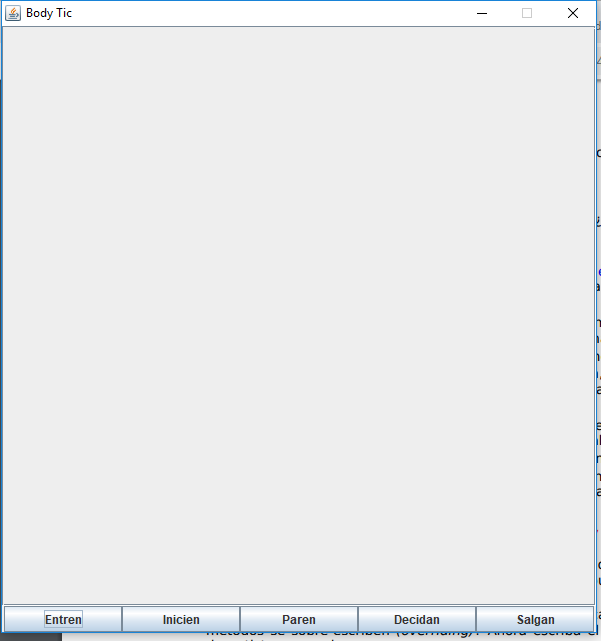


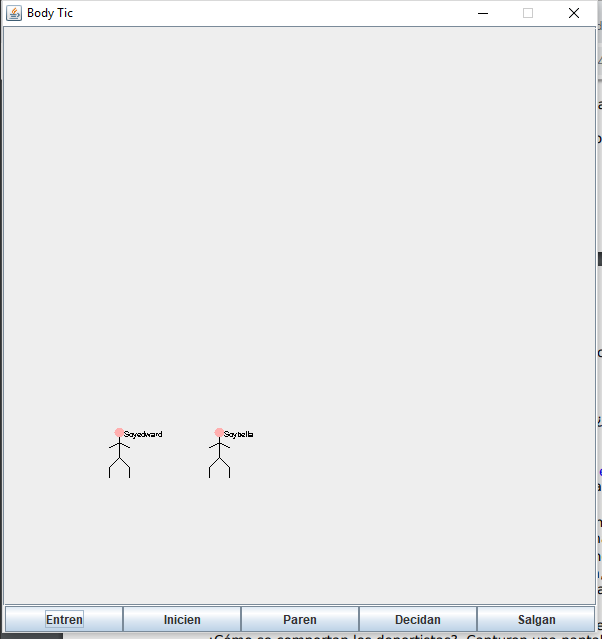
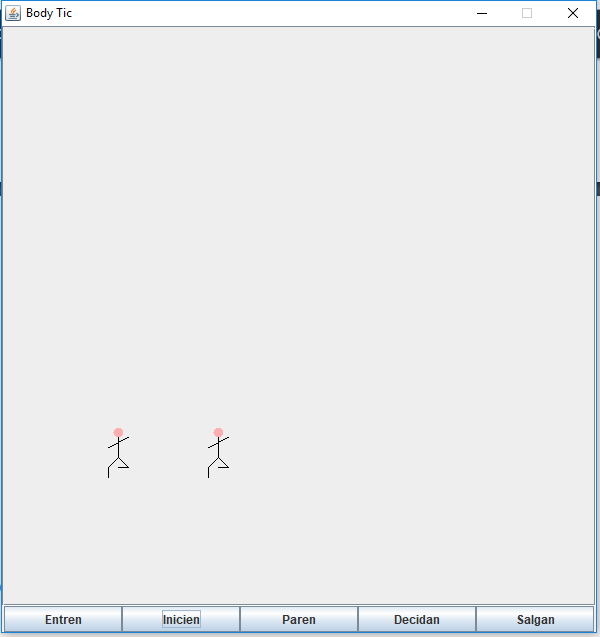
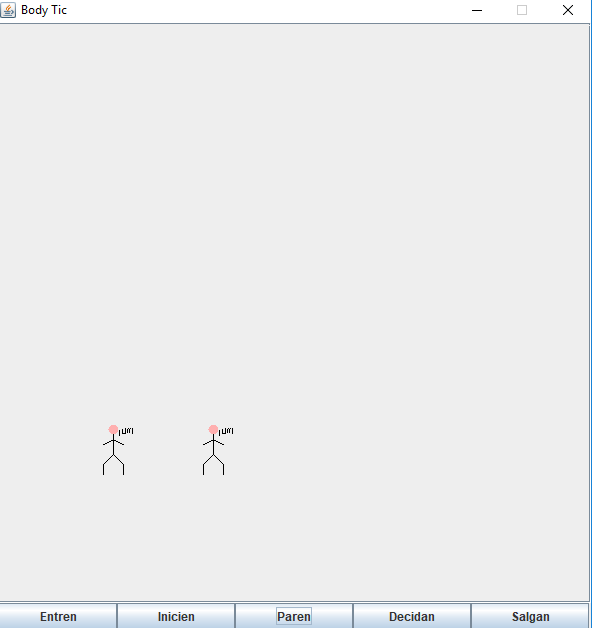
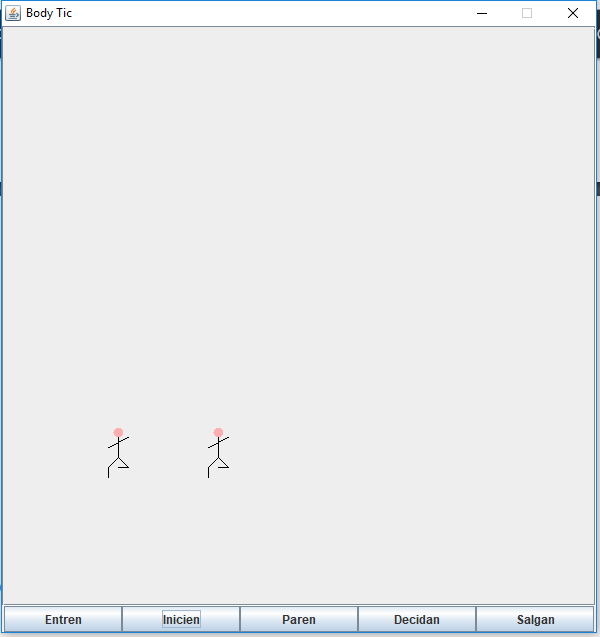
El contenido del directorio aplicación posee las clases y el archivo java del paquete aplicación, mientras que el directorio presentación posee las clases y los diversos archivos java que usa BodyTic. Su coincidencia es que ambos guardan esencialmente lo mismo.

**Arquitectura Detallada**

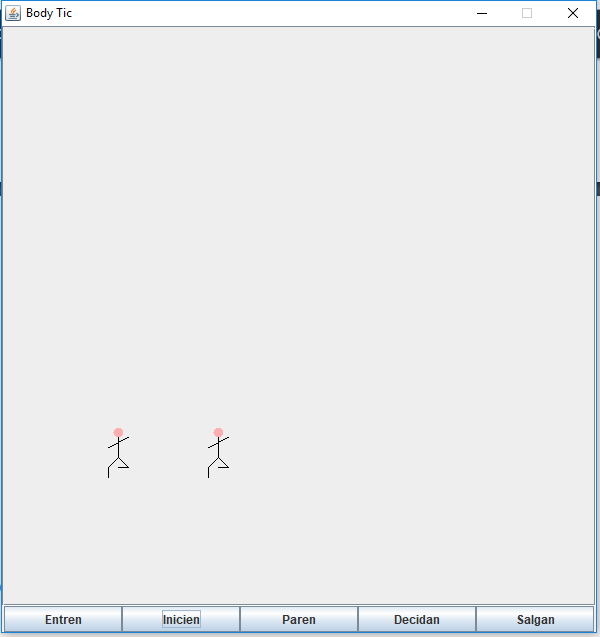
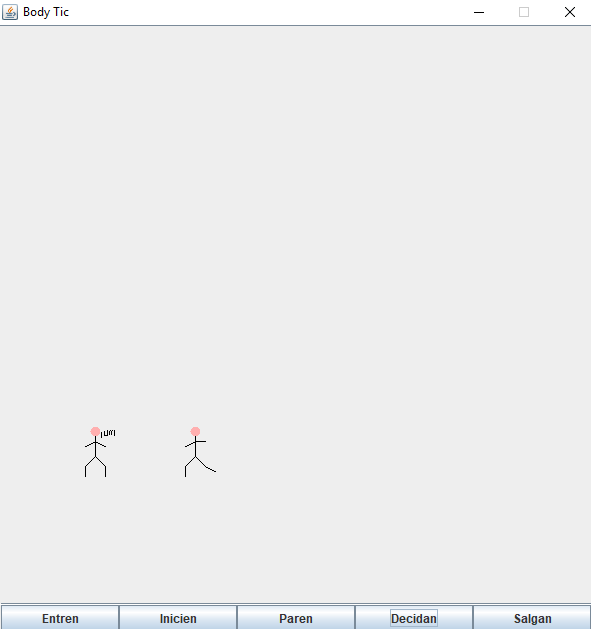
**Iniciando con los deportistas normales**

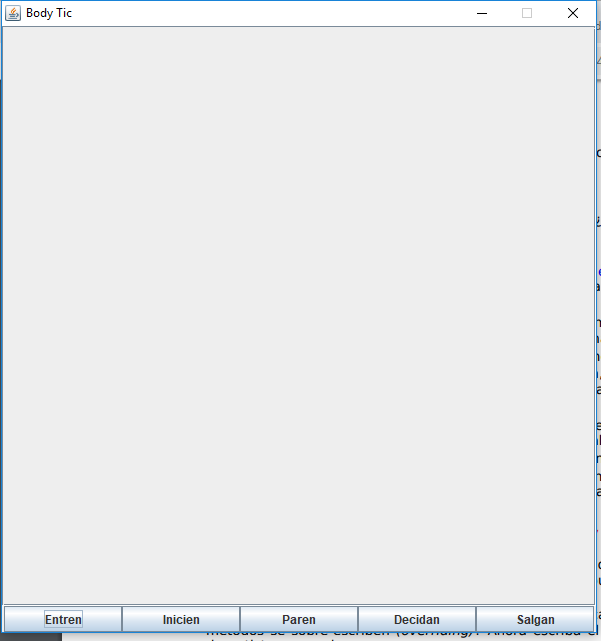
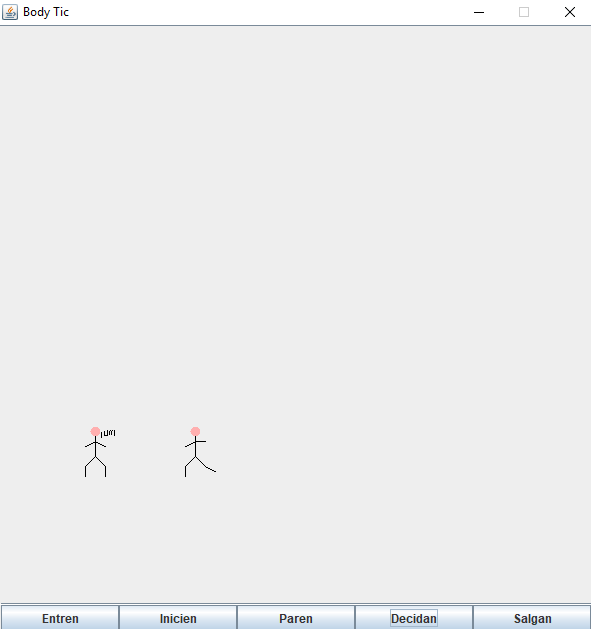
1. Una ArrayList, si puede recibir deportistas, porque la clase deportista implementa la interfaz EnSalon de la cual se compone el ArrayList.
2. La superclase Persona hereda a Deportista y la interfaz EnSalon implementa a Deportista.
3. Mover sus piernas, brazos y desplazarse. No pueden desplazarse ni moverse en una dirección concreta de forma distinta. No debería aprender a hacer nada más.
4. Sabe decidir si moverse o quedarse quieto. No puede definir su forma y mandar un mensaje vacío de forma distinta. Debe aprender cómo moverse, quedarse quieto y a reconocer su color y posición.
5. De color negro, se mueve como una persona dentro del salón, decide aleatoriamente y se inicia quieto como una persona.
6. Se ven reflejados en la posición inicial X y Y en que se inicializaron



1. Los deportistas mueven sus brazos y sus piernas. No hablan.
2. Los deportistas vuelven a la posición inicial que tenían cuando se ejecutó el método entrar.

9. Al ejecutar el método decidan los deportistas eligen aleatoriamente si paran o inician, como se puede ver reflejado en la imagen de abajo.



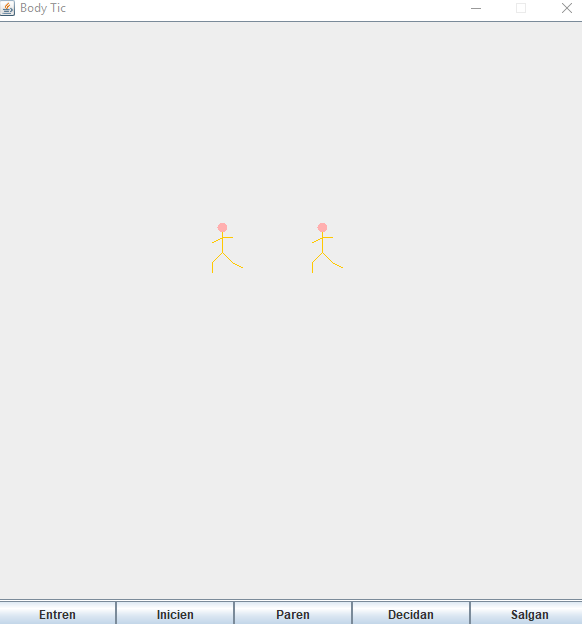
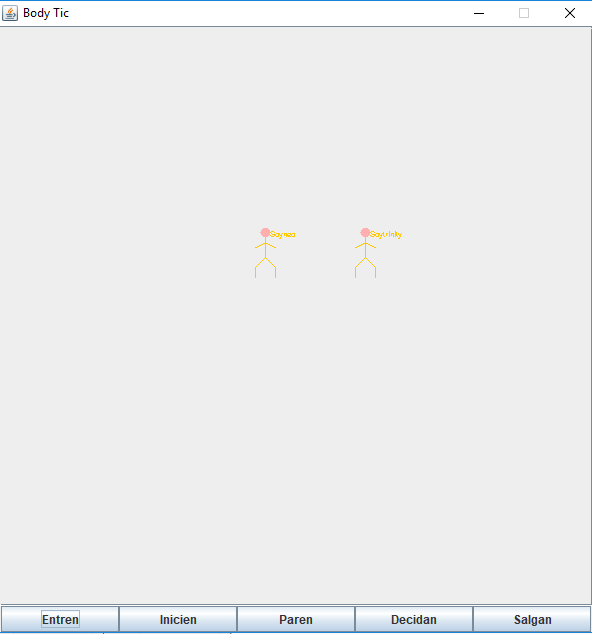
1. Al ejecutar el método salgan los deportistas ya no se ven reflejados en el salón.

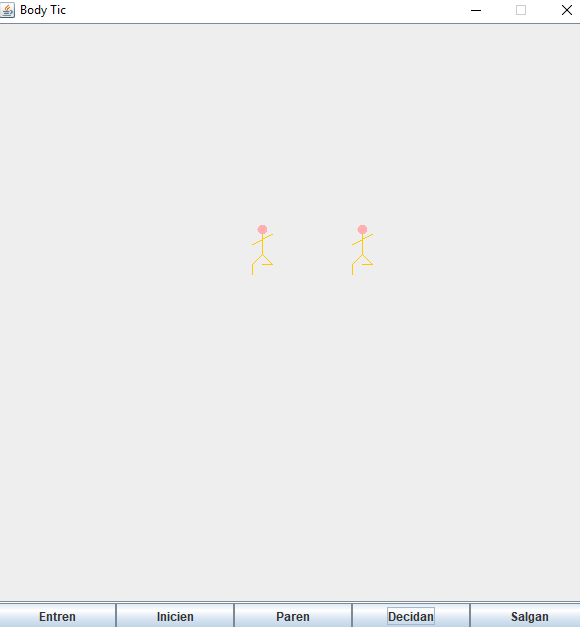
**Ciclo 2. Incluyendo a los deportistas avanzados**

**El objetivo de este punto es recibir en el salón deportistas avanzados. Los deportistas avanzados, vestidos de naranja, se mueven más, sólo paran después de dos órdenes y, cuando les piden que deciden, siempre hacen ejercicio.**

1.Se creó una nueva clase llamada DeportistaAvanzado que extiende a la clase Deportista y por lo tanto también se conoce como la Interface EnSalon. Se sobrescribieron los siguientes métodos de la clase Deportista: inicie (), pare () y decida ().

2. Los deportistas avanzados inicial como lo hacen normalmente los deportistas. Se mueven una mayor distancia a la que lo haría un deportista normal. Además, vemos que cuando se ejecutó el método decidir el héroe también se movió, puesto que siempre hace ejercicio. Luego, cuando se ejecutó el método pare el deportista efectivamente paró puesto que ya se habían ejecutado dos órdenes.





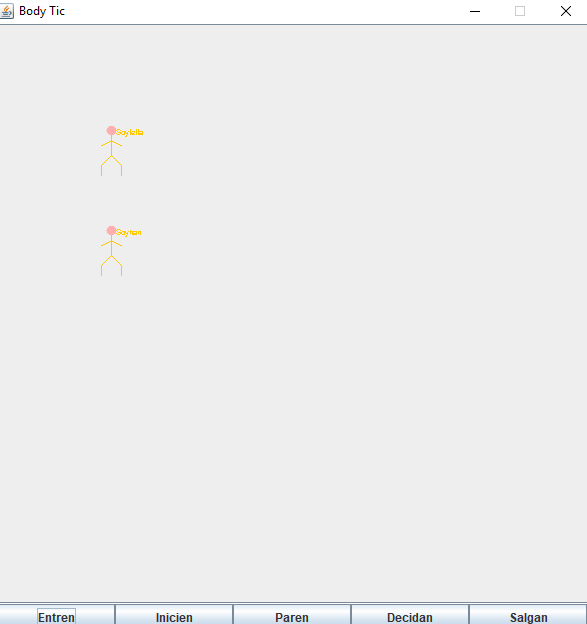
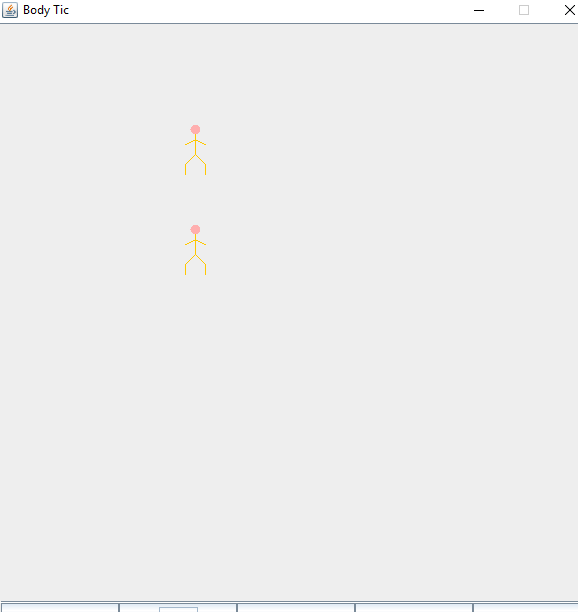
**Ciclo 2. Incluyendo a los deportistas habladores**

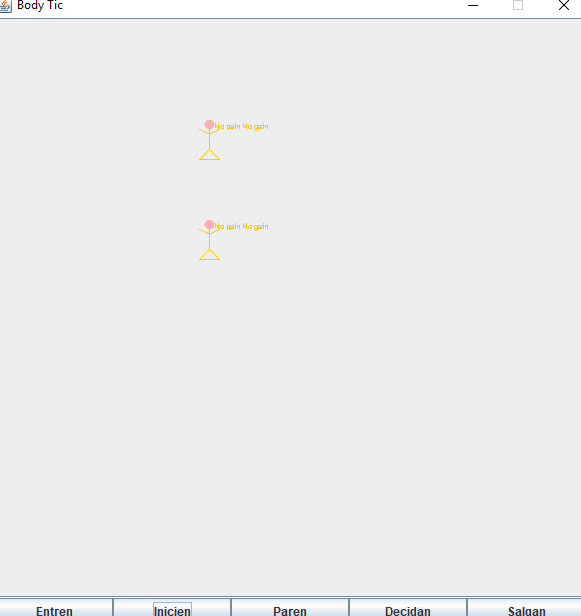
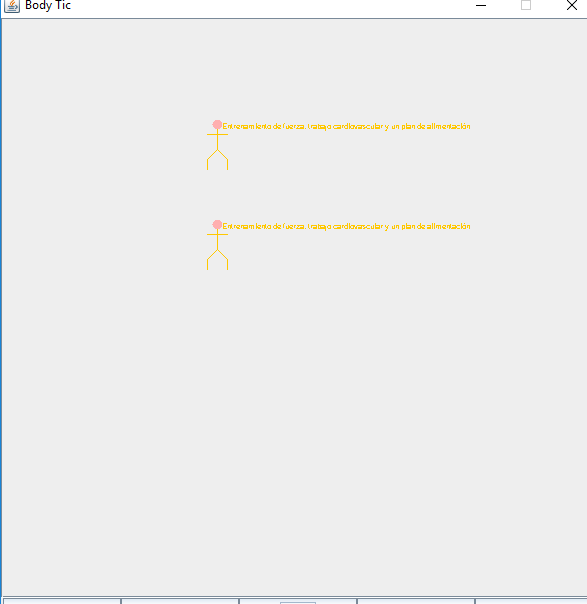
**El objetivo de este punto es recibir en el salón deportistas habladores. Los deportistas habladores están vestidos de naranja; al ejercitarse saltan (van subiendo los pies lentamente y luego los baja) y estando arriba se mueven a la derecha, y habla poco; y cuando descansan hablan mucho.**

1.Se crea una nueva clase llamada DeportistasHabladores que extienden la clase Deportista y por lo tanto implementan la Interface EnSalon. Se sobrescriben los siguientes métodos: inicie (), pare ()

2. Entren e Inicien: Los deportistas entran al salón e indican su nombre. Se preparan para saltar.

Inicien y Paren: Los deportistas saltan recogiendo sus piernas y muestran un pequeño mensaje. Cuando se ejecuta el método paren no lo hacen, siguen ejercitando mostrando un mensaje motivacional como se muestra en la imagen.





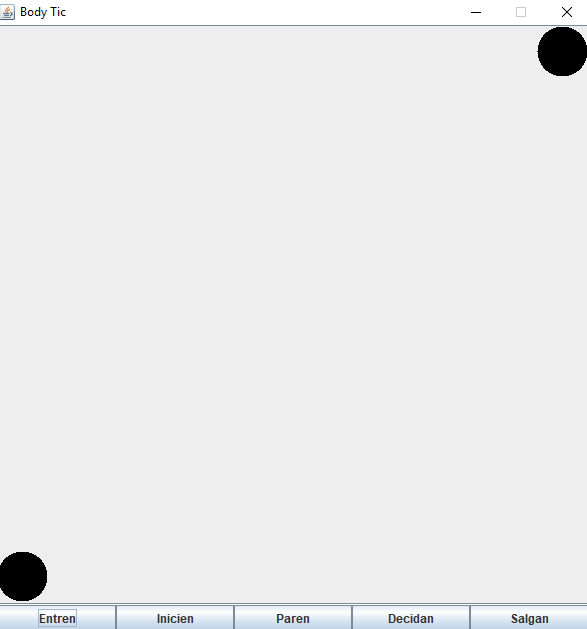
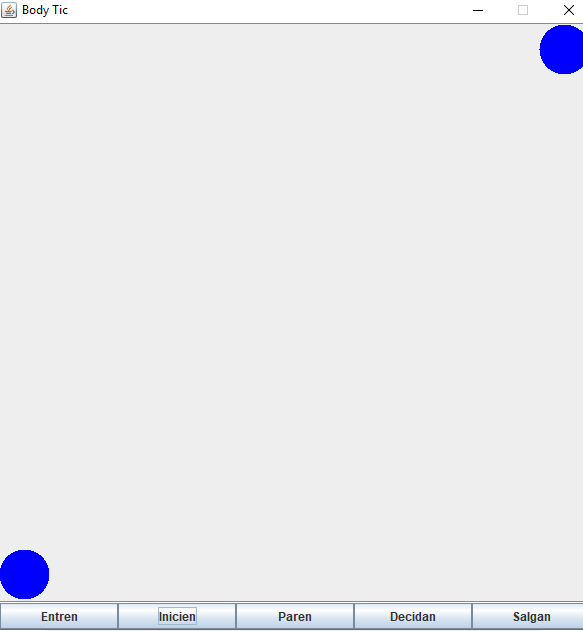
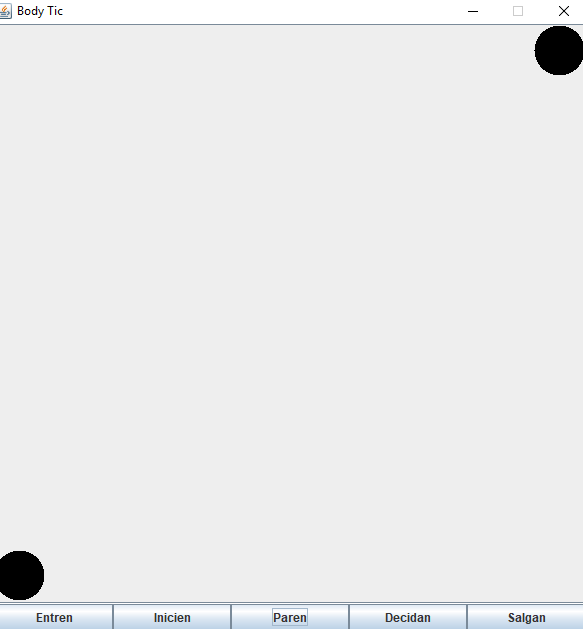
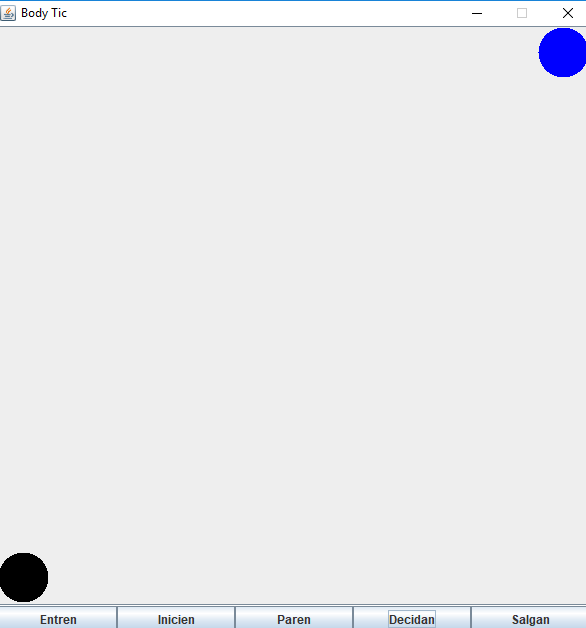
**Ciclo 3. Adicionando bolas de pilates**

**El objetivo de este punto es incluir en el Salón bolas de pilates especiales de colar (sólo vamos a permitir un tipo de bolas). Las bolas se encienden cuando inician, se apagan cuando paran y decien de manera estándar.**

1.Es necesario crear una nueva clase llama ‘Bola’ que implemente la interface EnSalon. Además, en la Interfaz EnSalon es necesario agregar a la lista de bolas la posición “Bola” para que la clase BodyTicGUI logre reconocerla y dibujarla. Se sobrescriben los siguientes métodos: forma (), pare (), inicie () getColor (), getPosicionY (), getPosicionX ().

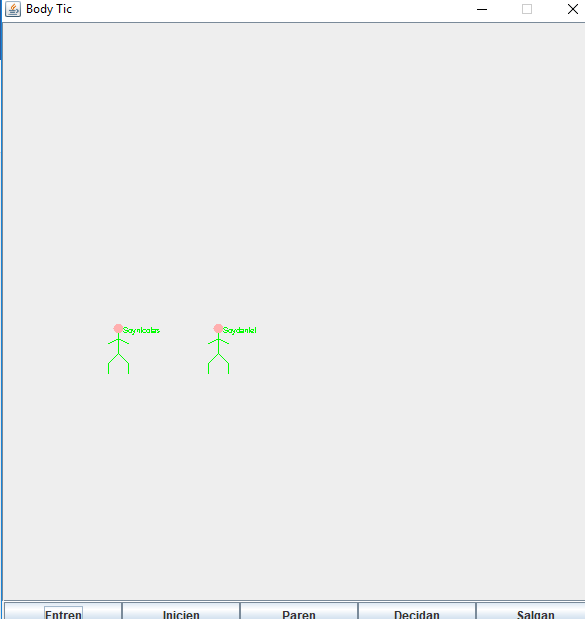
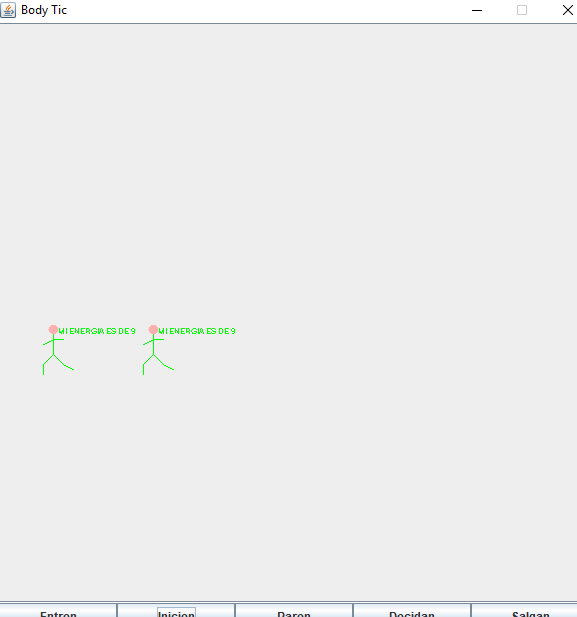
2. No. Se debe procurar únicamente que la nueva clase creada implemente la Interface EnSalon.

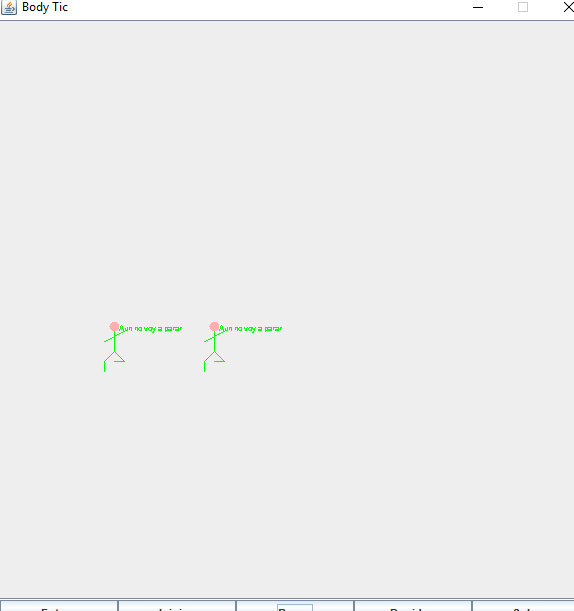
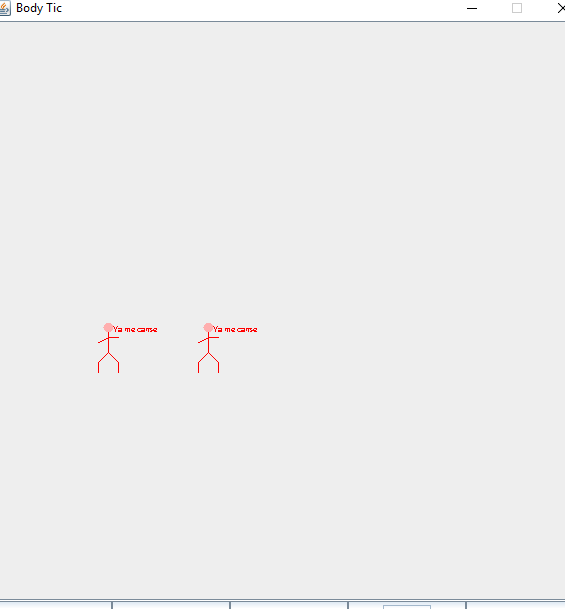
3 y 4. Al iniciar las bolas se ubican en el suroeste y el noroeste del salón. Cuando inician cambiar a color azul y cuando paran a color negro. El método decida elige aleatoriamente entre los dos procesos anteriores.



**Ciclo 4. Nueva Deportista: Proponiendo y diseñando El objetivo de este punto es permitir recibir en un nuevo tipo de deportista**

**1.Deportistas Vigorexo**: Es un tipo de deportista que se ejercita de acuerdo a la energía que tenga. Una vez agote su energía el deportista no puede volver a ejercitarse. El usuario define la energía inicial del deportista. Durante el ejercicio el deportista informa qué energía tiene. Si se le pide parar antes de que acabe su energía no lo hará.

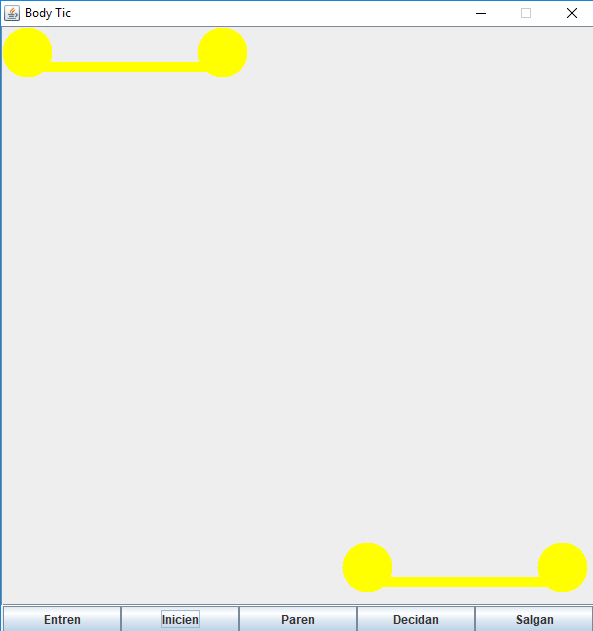
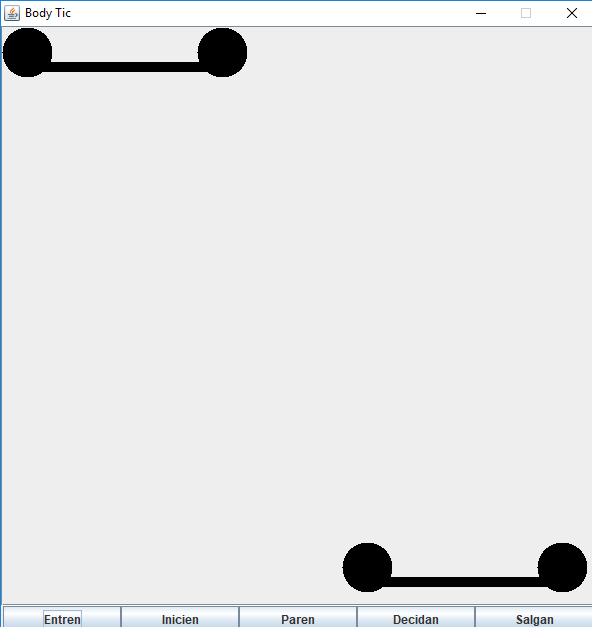
**Entren e Inicio:** Los deportistas dicen su nombre e informan de la energía que poseen

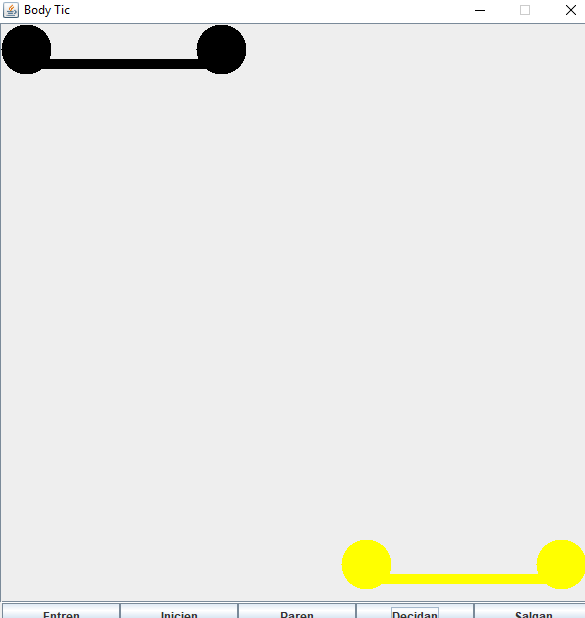
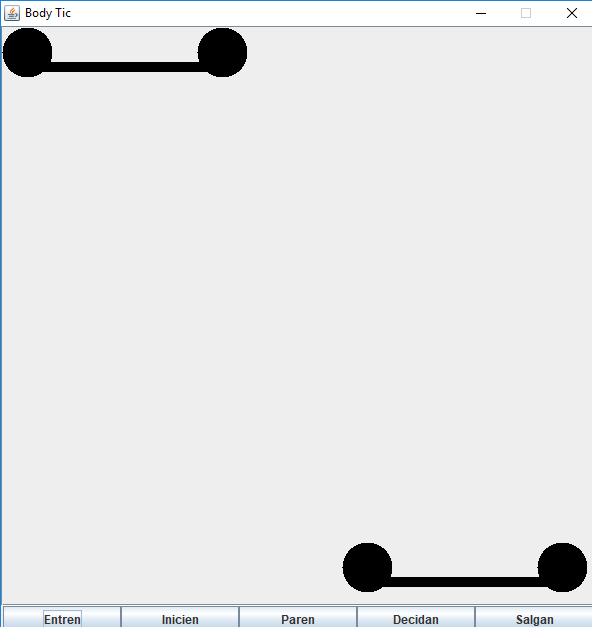
**Paren: Si se le pide al deportista que pare antes de que se le acabe la energía no podrá hacerlo y dirá “Aun no puedo parar”. Por el contrario, cuando se le acabé la energía se pondrá de color rojo y dirá “Ya me cansé”.**

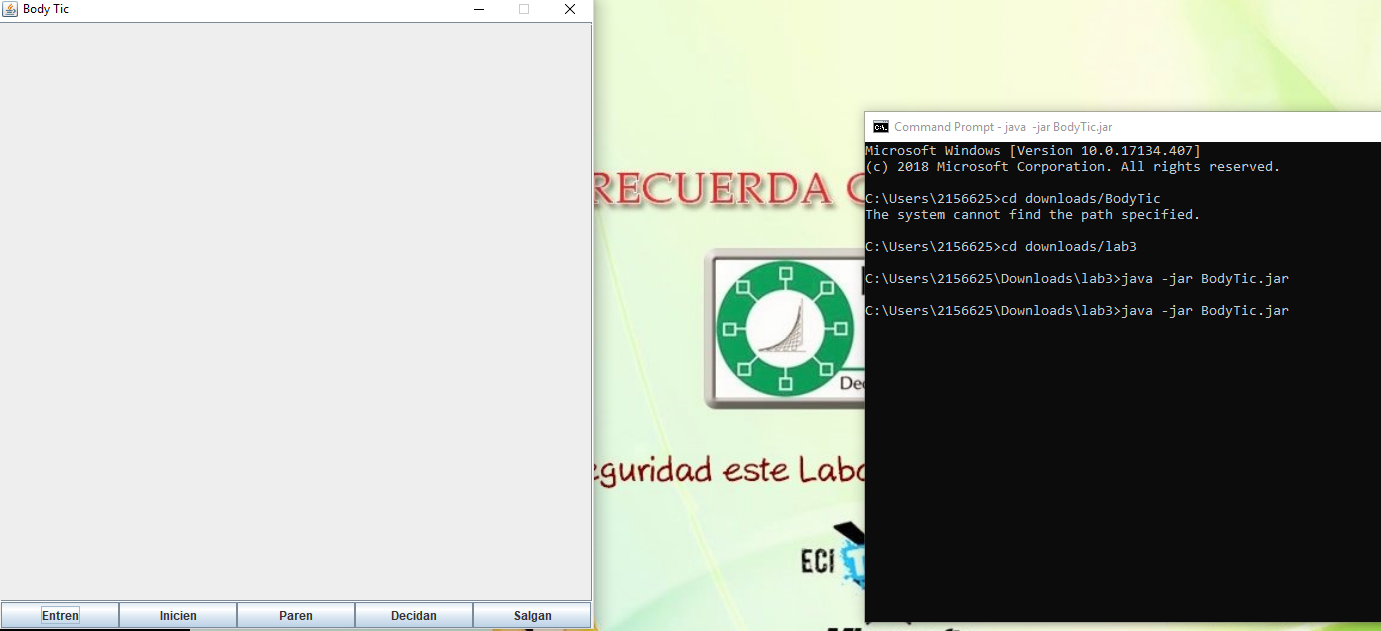
**Ciclo 5. Nuevo elemento: Proponiendo y diseñando El objetivo de este punto es permitir recibir en un nuevo elemento (no deportista) en el Salón.**

1.Se implementan las pesas del gimnasio. Para eso se crea una clase llamada Pesa que implementa la Interface EnSalon. Además, se modifica la lista de Figuras de tal manera que sea posible dibujar la pesa. Las pesas constan de dos círculos y un rectángulo que representan su forma. Las pesas cambian de color cuando inician y vuelven a su color inicial cuando deciden parar.

2. Inicien y paren: Se dibujan las pesas de acuerdo a un tamaño elegido por el usuario y se ubican en las posiciones que este desee. Al iniciar cambian al color amarillo.



**Deciden y paren**: Al decidir deciden aleatoriamente entre cambiar de color o no. Cuando paran vuelven al color elegido inicialmente.



Las ventajas es que se puede entregar de forma comprimida, en un solo archivo y este fácilmente es ejecutable sin que el cliente sea capaz de ver el código, además de que este es fácilmente portable a otras distribuciones de software debido a que solamente es necesario que este instalada la JVM (Java Virtual Machine)

**DE BLUEJ A CONSOLA**

**Comandos básicos del sistema operativo**

1. **Investiguen los comandos para moverse en la estructura de directorios: crear, borrar, listar su contenido y copiar o eliminar un archivo.**

DIR: Permite ver el contenido de un directorio

MD o MKDIR: Permite crear directorios

COPY: crear archivos

TREE: ver el árbol de directorios

XCOPY: Copiar directorios

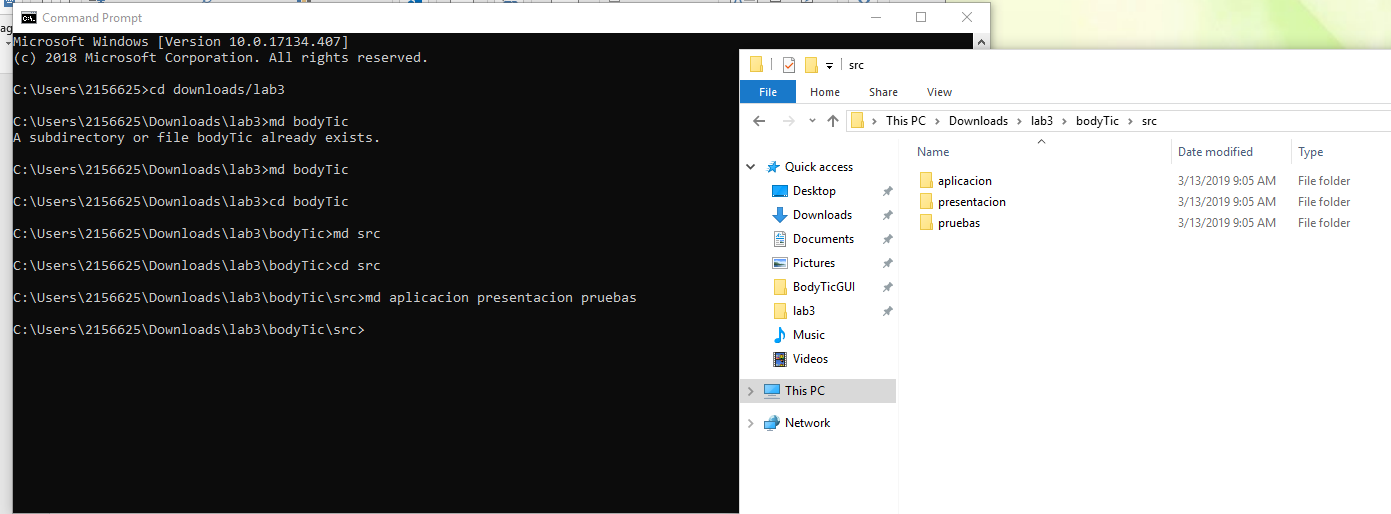
REN o RENAME: Cambiar nombre de directorio y archivos

MOVE: mover archivos y directorios

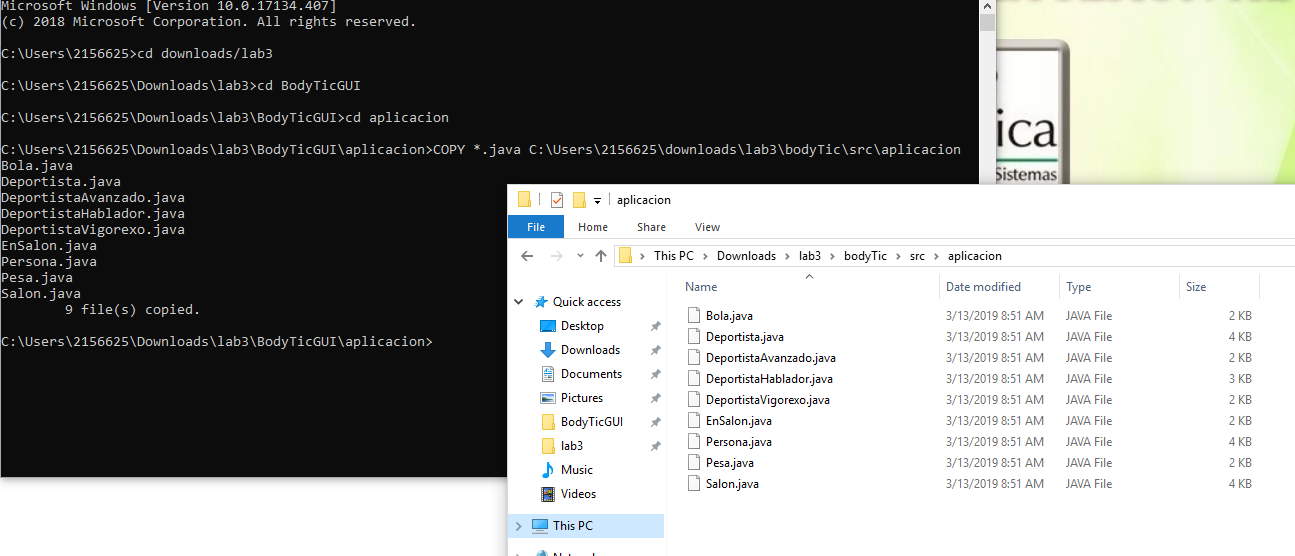
DEL: Eliminar archivos

RD: Eliminar directorios

1. **Organicen un nuevo directorio con la estructura propuesta para probar desde allí su habilidad con los comandos de consola. Consulten y Capturen el contenido de su directorio**



1. **En el directorio copien únicamente los archivos \*.java del paquete de aplicación. Consulte y Capturen el contenido de src/aplicación**



**Estructura de proyectos java**

1. **Investiguen los archivos que deben quedar en cada una de esas carpetas y la organización interna de cada una de ellas.**

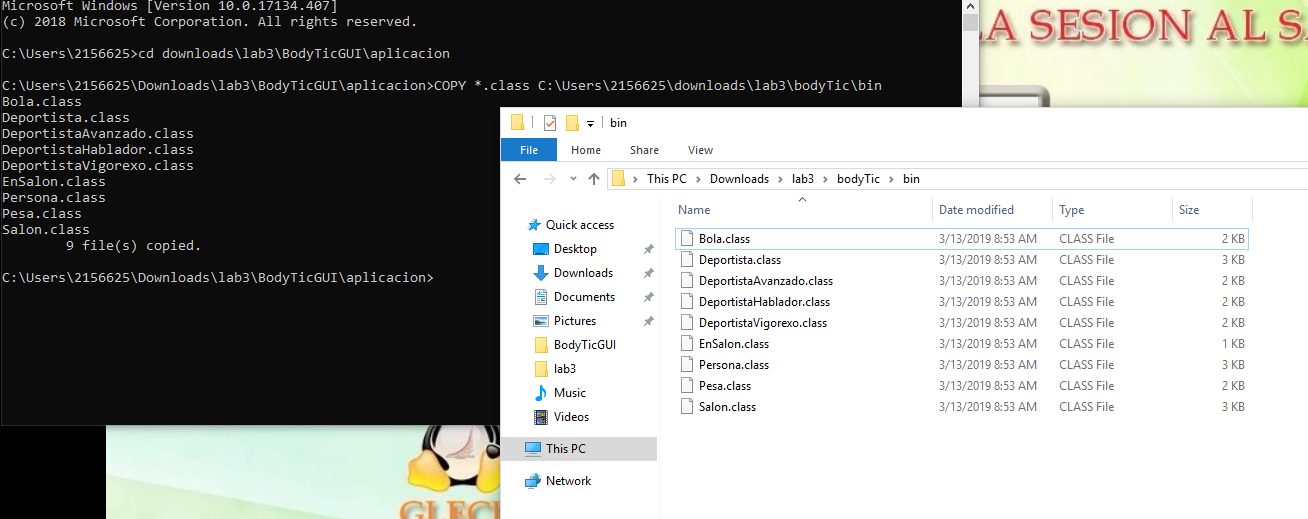
En src deben estar los códigos fuentes .java

En bin deben estar los bytecodes de los códigos fuentes.

En docs. se encuentra la documentación del proyecto

1. **¿Qué archivos debería copiar del proyecto original al directorio bin? ¿Por qué? Cópielos y consulte y Capturen el contenido del directorio que modificó.**

Los archivos .class porque contienen el bytecode del Proyecto.



**Comandos de java**

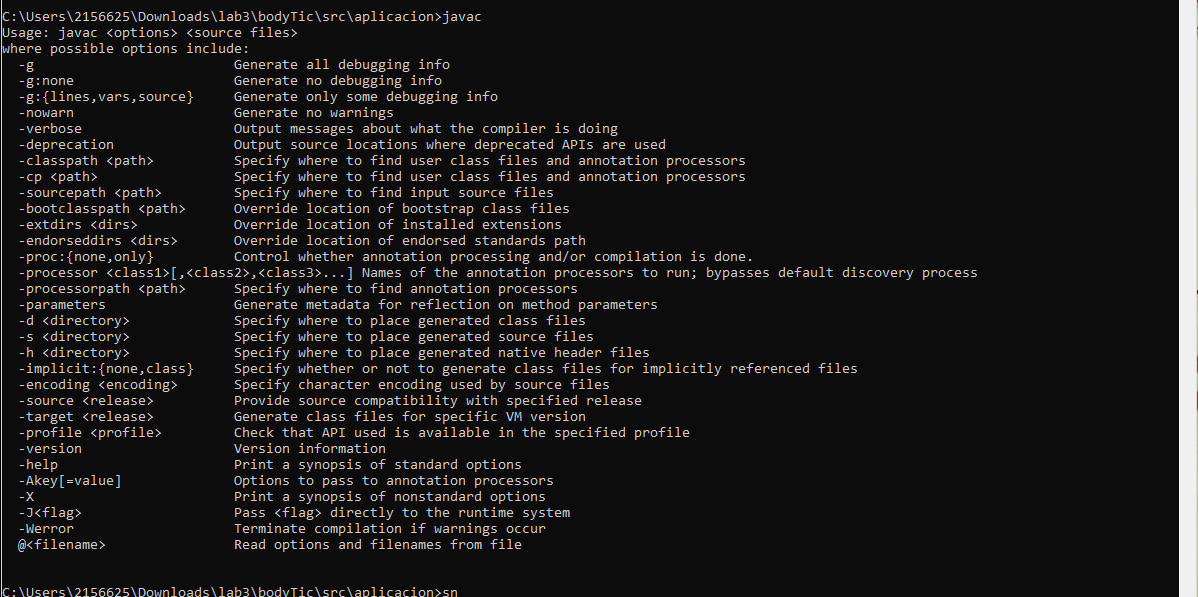
1. **Consulte para qué sirven cada uno de los siguientes comandos: javac java javadoc jar**

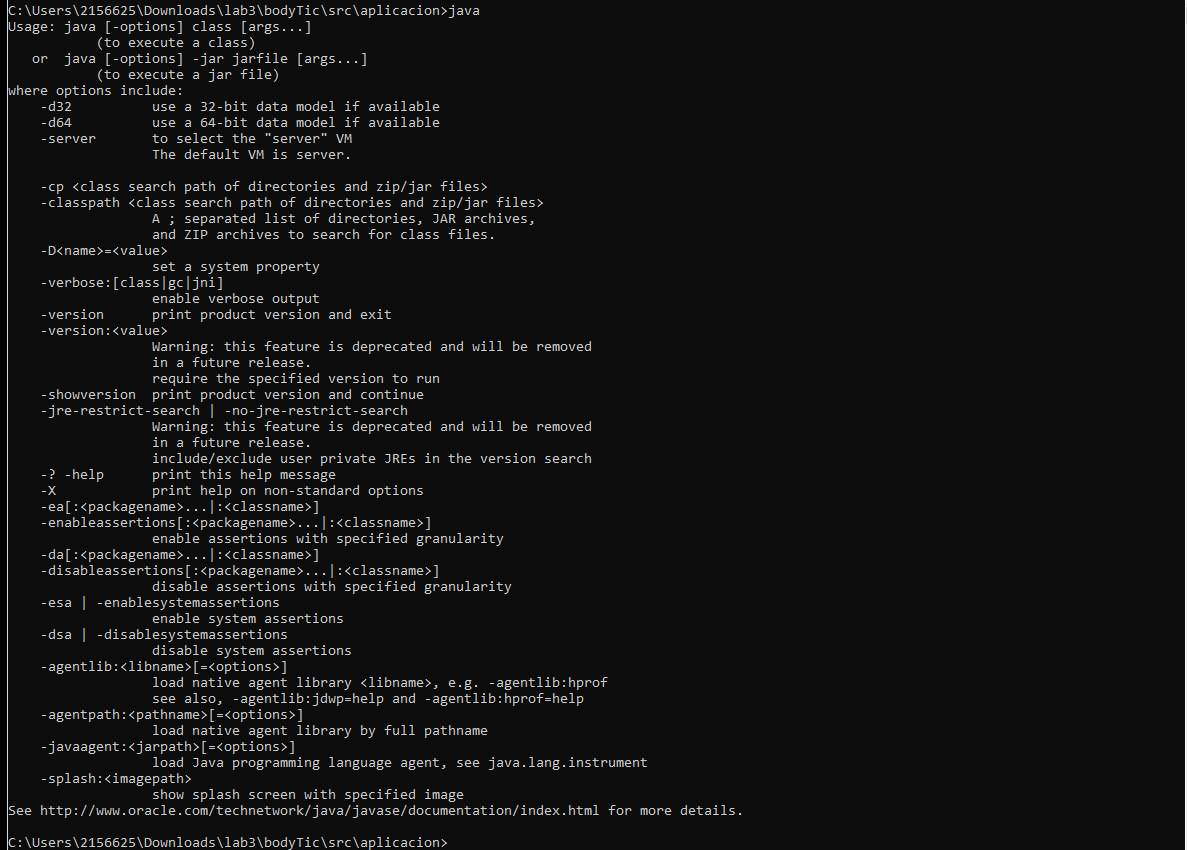
Javac: Compila los archivos con código fuente (.java) en byte-codes Java (.class)

Java: Es el intérprete de Java; es un componente de la JVM capaz de interpretar los archivos (.class) en la computadora

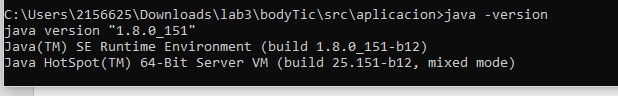
Javadoc: Genera páginas de HTML de API de la documentación de los códigos fuente .java

Jar: Comprime proyectos java en un solo archivo y también ejecuta estos archivos.

1. **Cree una sesión de consola y consulte en línea las opciones de los comandos java y javac. Capturen las pantallas.**



1. **Busque la opción que sirve para conocer la versión a que corresponden estos dos comandos. Documente el resultado.**

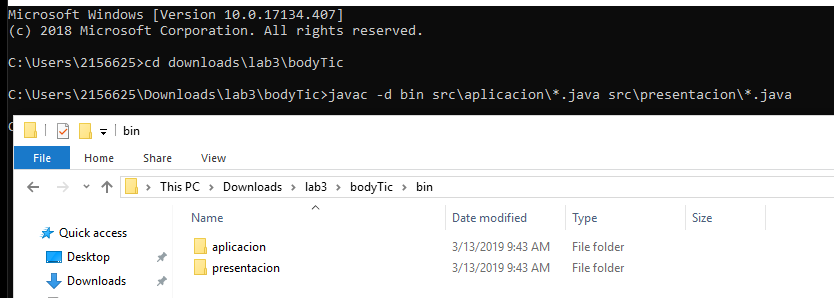


Resultado: Version 1.8.0\_151

**Compilando**

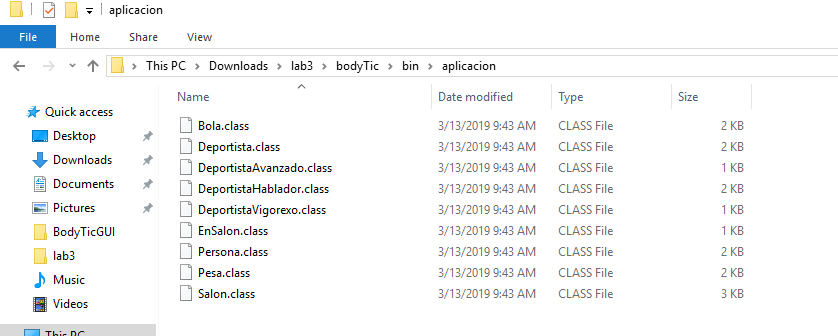
1. **Utilizando el comando javac, desde el directorio raíz (desde bodyTic con una sola instrucción), compile el proyecto. ¿Qué instrucción completa tuvo que dar a la consola para compilar TODO el proyecto? Tenga presente que se pide un único comando y que los archivos compilados deben quedar en los directorios respectivos.**

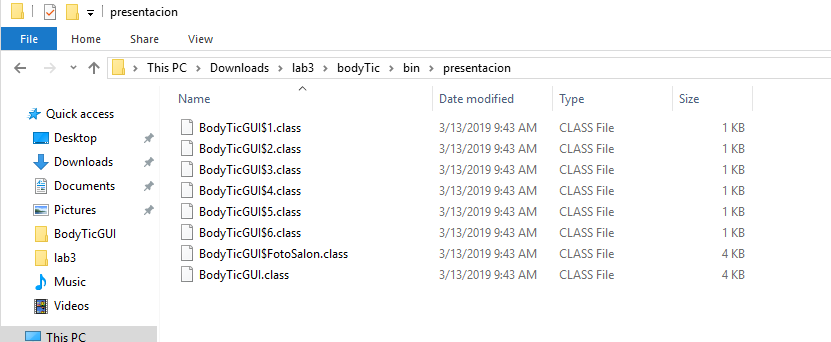
Para compilar todo: javac -d bin src\aplicación\\*.java src\presentación\\*.java



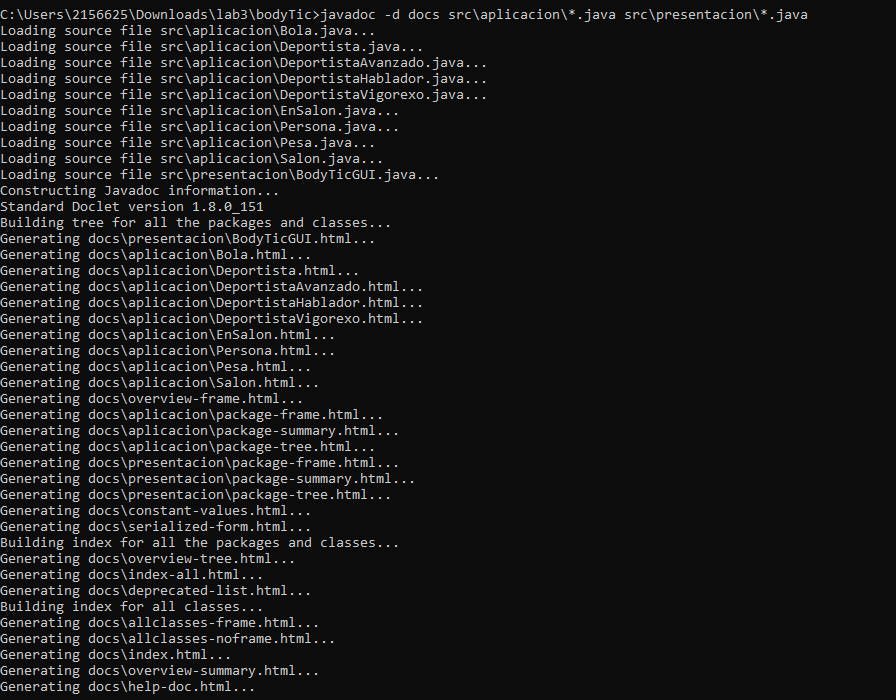
1. **Revise de nuevo el contenido del directorio de trabajo y sus subdirectorios. ¿Cuáles nuevos archivos aparecen ahora y dónde se ubican?**

Aparecieron nuevos archivos .class en los directorios bin de la aplicación y la presentación.



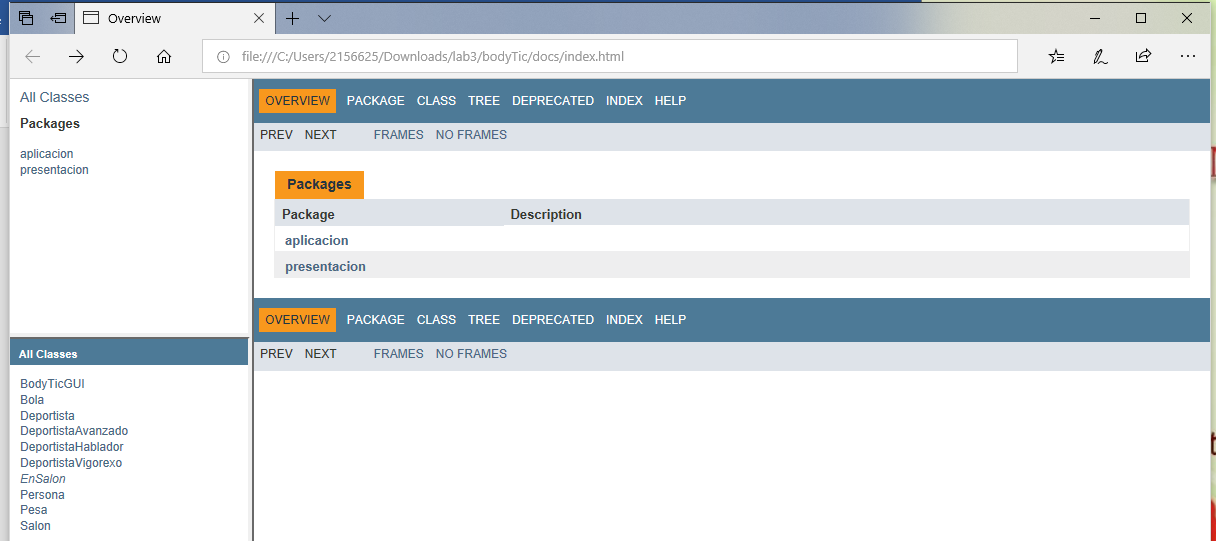


**Documentando**

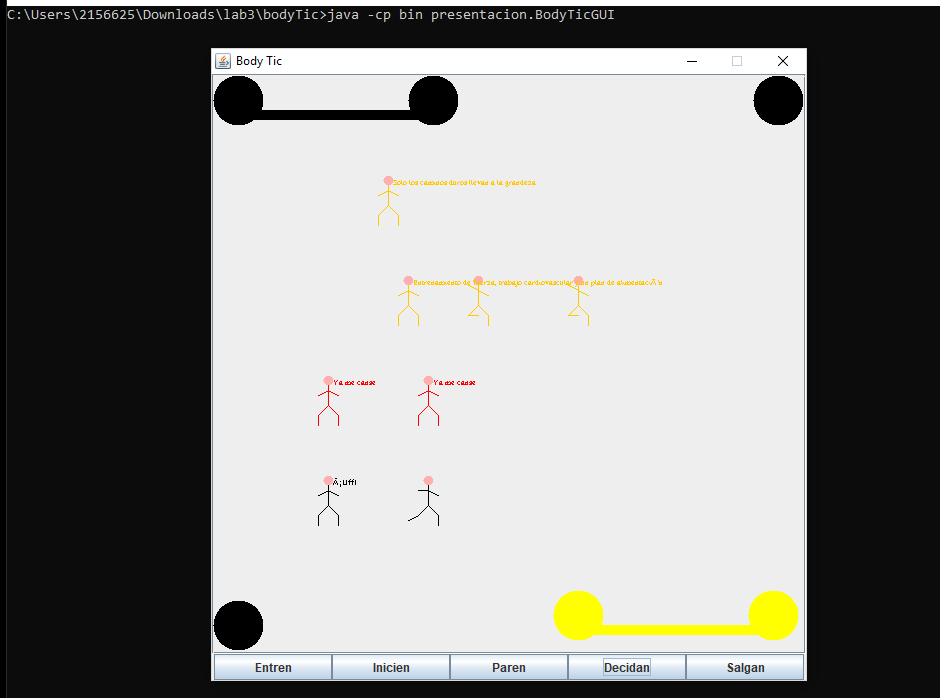
**1. Utilizando el comando javadoc, desde el directorio raíz, genere la documentación (API) en formato HTML, en este directorio. ¿cuál es el comando completo para generar esta documentación?**

1. **¿Cuál archivo hay que abrir para empezar a navegar por la documentación? Ábralo y Capturen la pantalla.**

El archivo index.html



**Ejecutando**

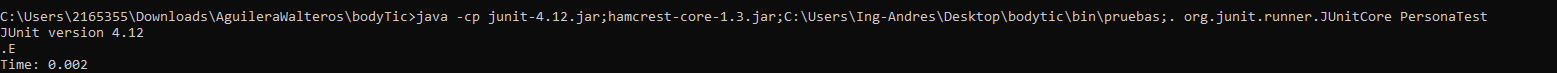
1. Empleando el comando javo, desde el directorio raíz, ejecuten el programa. ¿Cómo utilizó este comando?

**Probando**

1.Adicionen ahora los archivos del directorio pruebas y trate de compilar nuevamente el programa.



2.Tenga en cuenta que estas clases requieren la librería junit 4.8. ¿Cómo se incluye un paquete para compilar? ¿Qué instrucción completa tuvo que dar a la consola para compilar? 2. ejecuten desde consola las pruebas . ¿Cómo utilizó este comando?. Puede ver ejemplos de cómo ejecutar el“test runner”en: http://junit.sourceforge.net/doc/cookbook/cookbook.htm 3. Pegue en su documento el resultado de las pruebas



**Empaquetando**

1. Consulten como utilizar desde consola el comando jar para empaquetar su programa entregable en un archivo .jar, que contenga los archivos bytecode necesarios (no las fuentes ni las clases de prueba), y que se pueda ejecutar al instalarlo en cualquier directorio, con solo tener la máquina virtual de java y su entorno de ejecución (JRE).



2. ¿Cómo se ejecuta el proyecto empaquetado?