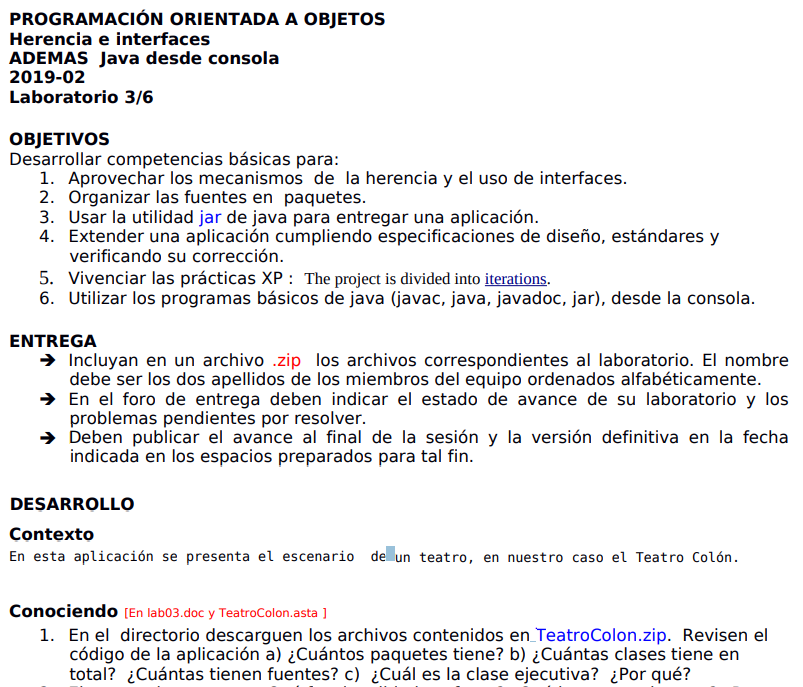
**Juan Romero – Sebastián Villamarín R**

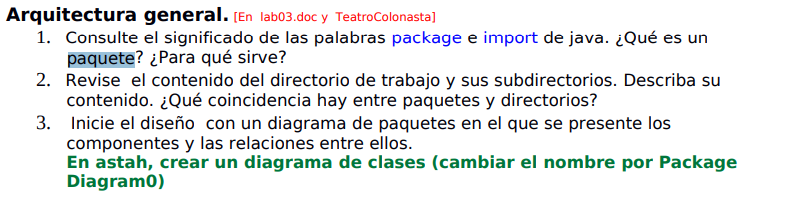


1. Tiene 3 paquetes
2. En total tiene 6 clases.
3. La clase ejecutiva es TeatroColonGUI porque esta es la que abre las demás clases.

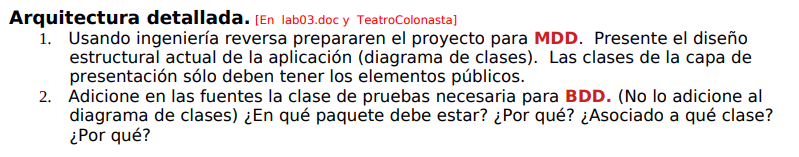


El paquete aplicacion Tiene como clases enEscena, Teatro que tiene 2 funcionalidades que son demeTeatro() y cambieTeatro(), Persona tiene como funcionalidad es aplicacion.Persona(nombre,xposition,yposition), Luz tiene como funcionalidad es aplicacion.luz(), Actor tiene como funcionalidad es aplicacion.Actor(teatro,string,int,int), pero en si no realizan ninguna acción de valor debido a que faltan metodos por implementar y complementar.

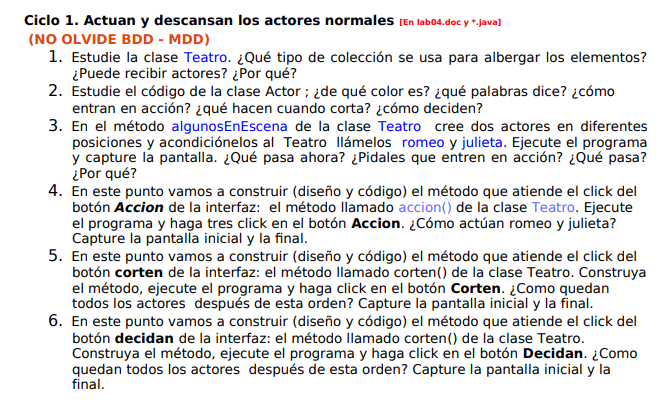
El paquete presentacion tiene como clase TeatroColonGUI que tiene como funcionalidades main(String[] args), paintComponent(Graphics g), void humano(Graphics g, Persona e,int x, int y) pero en si no realizan ninguna acción de valor debido a que faltan metodos por implementar y complementar.



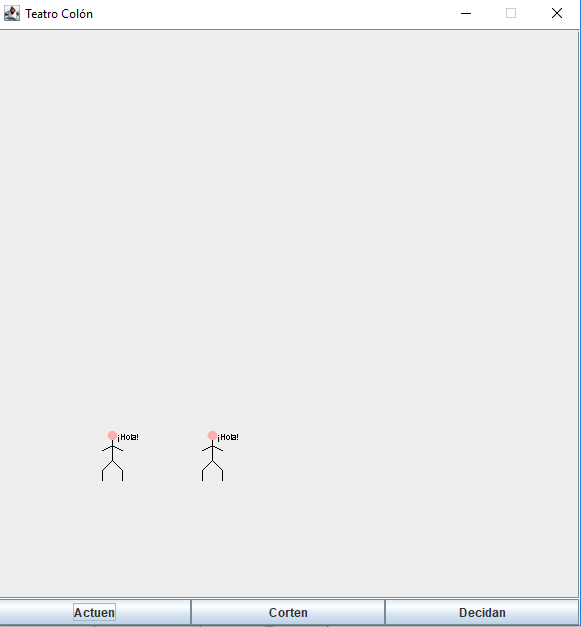
1. Un ejemplo de paquete es la aplicación donde cada una de sus clases debe tener “package aplicacion” y el import pone a disposición tosas las clases especificadas “java.awt.Color”, aca se importa solo la clase color del paquete awt.
2. Los paquetes son una forma de organizar grupos de clases. Un paquete contiene un conjunto de clases relacionadas bien por finalidad, por ámbito o por herencia. Los paquetes resuelven el problema del conflicto entre los nombres de las clases. Al crecer el número de clases crece la probabilidad de designar con el mismo nombre a dos clases diferentes. Las clases tienen ciertos privilegios de acceso a los miembros dato y a las funciones miembro de otras clases dentro de un mismo paquete. Con el import se hace el llamado de una clase que se encuentra en otro paquete o en sí mismo.
3. Que los paquetes se generan a partir de directorios y hay algunos directorios que necesitan de otros paquetes



2. Hay una clase que no está relacionada con nadie, los métodos sin terminar de implementar.

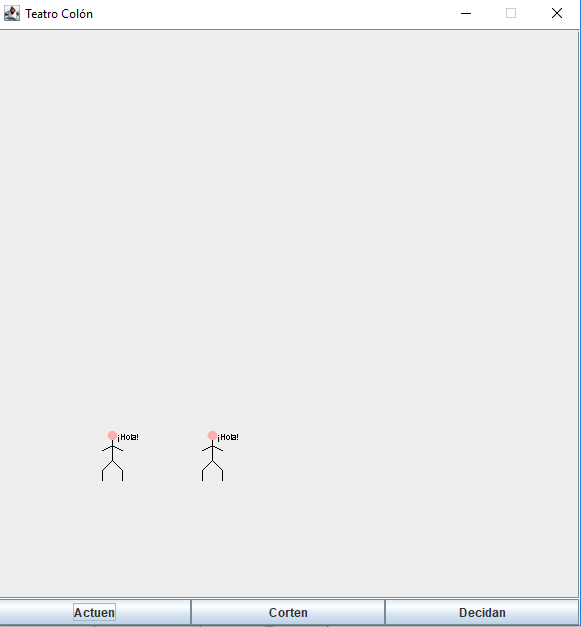


1. Usa una arraylist de tipo <en Escena>, donde puede albergar escenas ya que estas son elementos.
2. Es de color negro, la palabra que dice es ¡Hola!, entra en acción cuando en el metodo acute() entra en alguno de los condicionales ya que eso implica que hay partes del cuerpo, cuando le hacen corte bajan los brazon y las piernas en el piso y sin palabras
3. Al ejecutar el programa se crean los actores romeo y julieta y al darles que entren en acción no hacen nada ya que el método de acción no esta implementado.

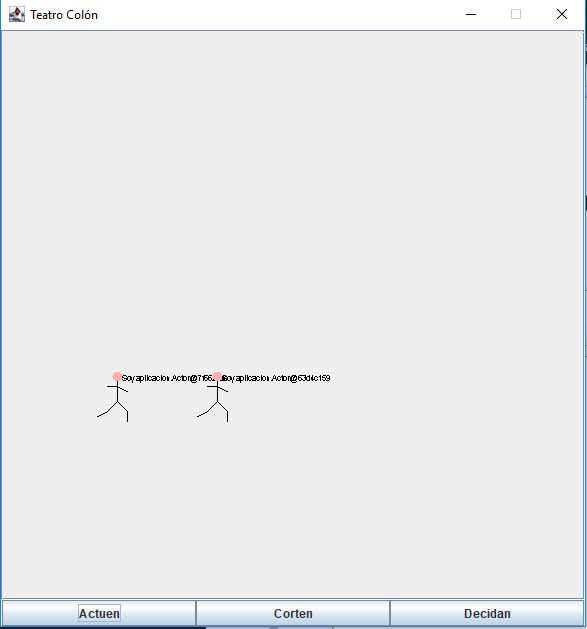


1. Al dar los clicks los actores realizan múltiples movimientos tanto con sus brazos y piernas. También se desplazan verticalmente y mandan mensajes.

Antes :

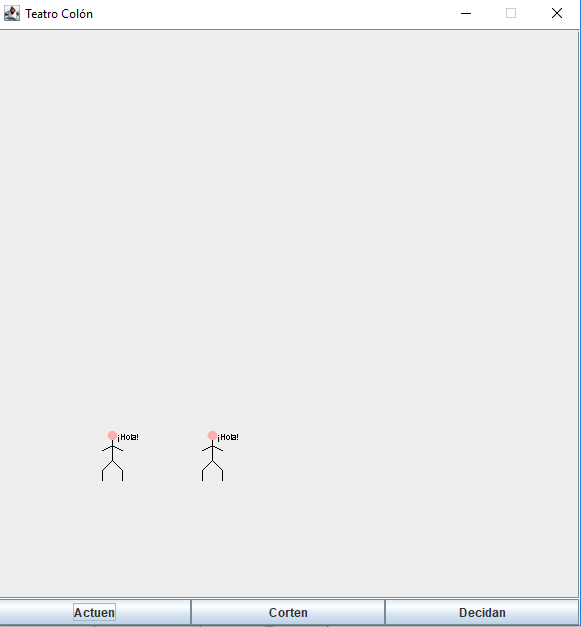


Después (3 clicks):

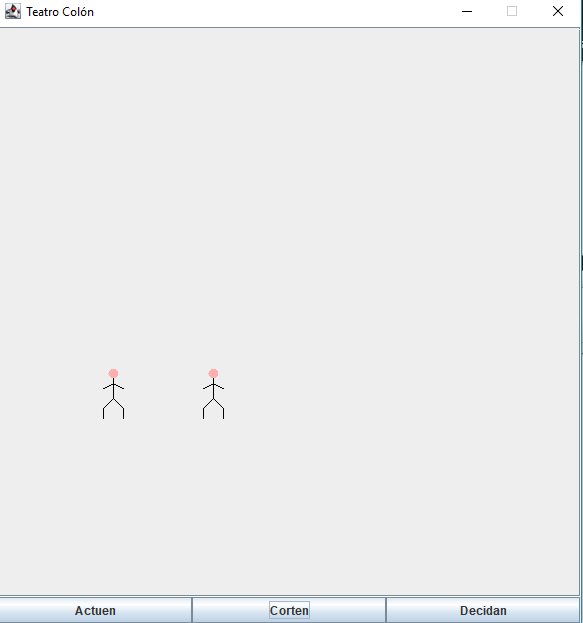


1. Al dar el botón de corte los actores quedan sin mensaje y tanto piernas como brazos abajo

Antes :

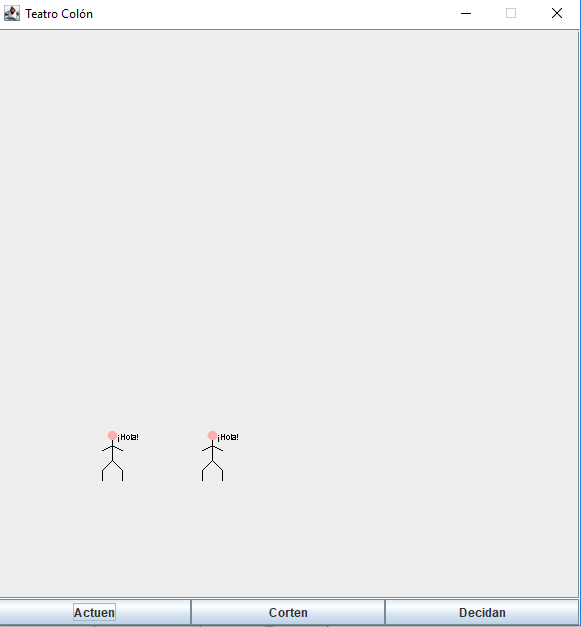


Después :

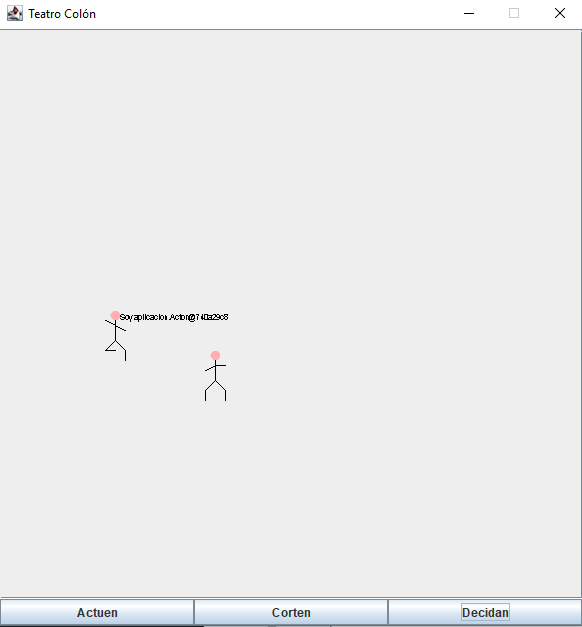


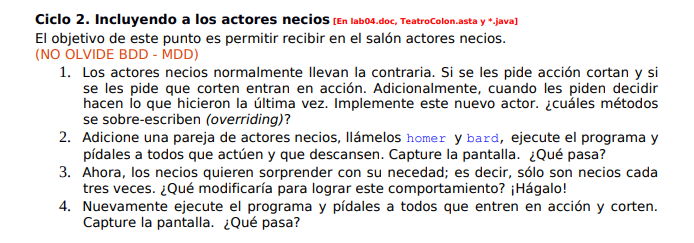
1. Los actores realizan de forma aleatoria acciones o cortes

Antes :



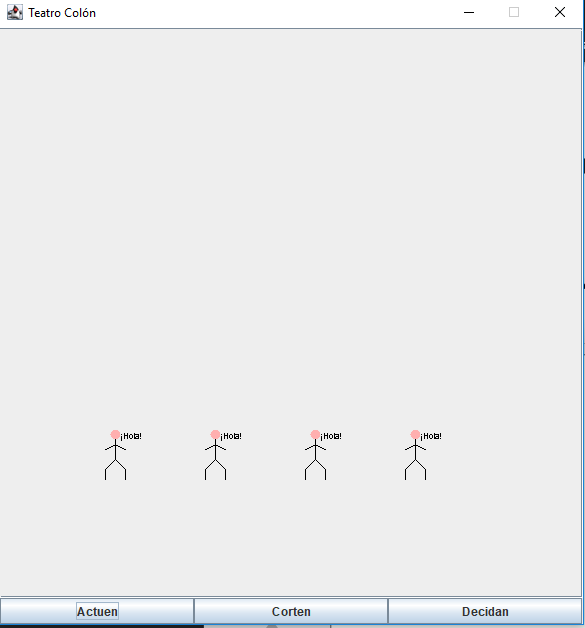
Despues:



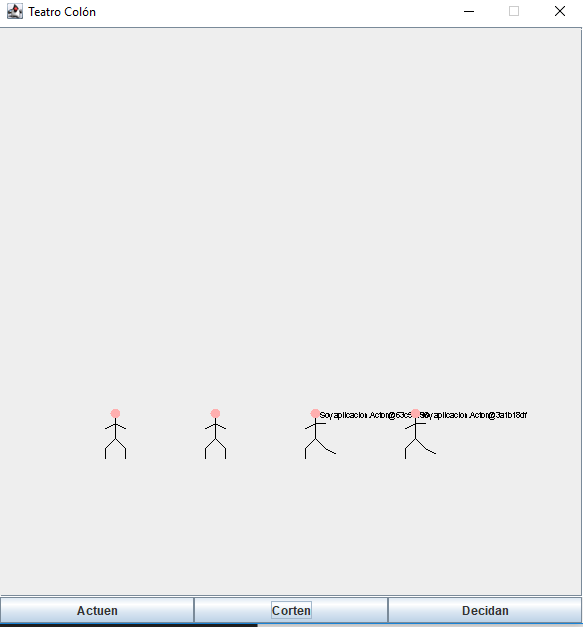


1. Se sobre – escribieron los métodos acción(), corte() y decidan()
2. Romero y julieta actúan y cortan normal a cambio homer y bard realizan primero el corte y luego la acción

Antes :



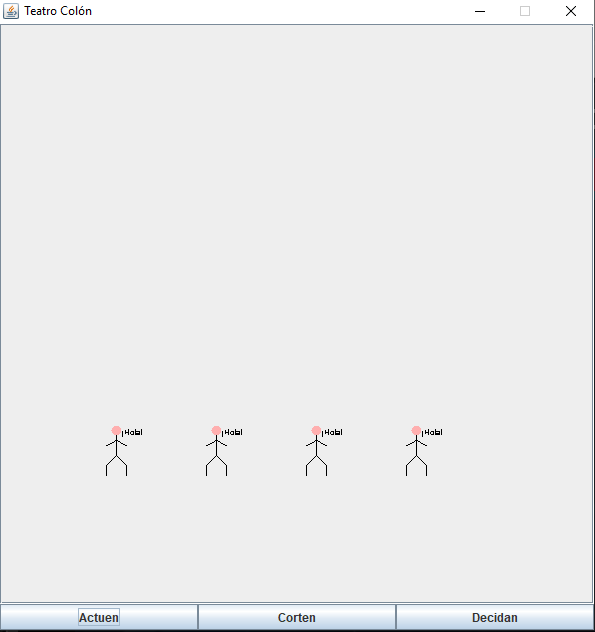
Después :



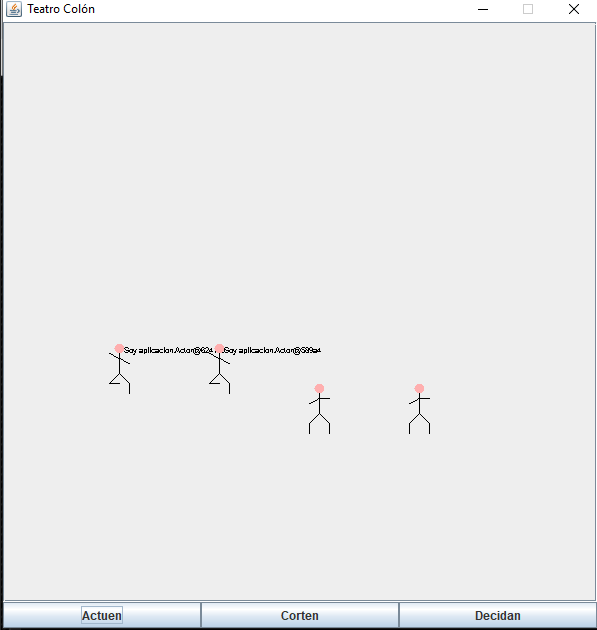
1. Tanto en los métodos acción() como corte() se les lleva un contador para que al tercer movimiento sean necios

En acción:

Antes :

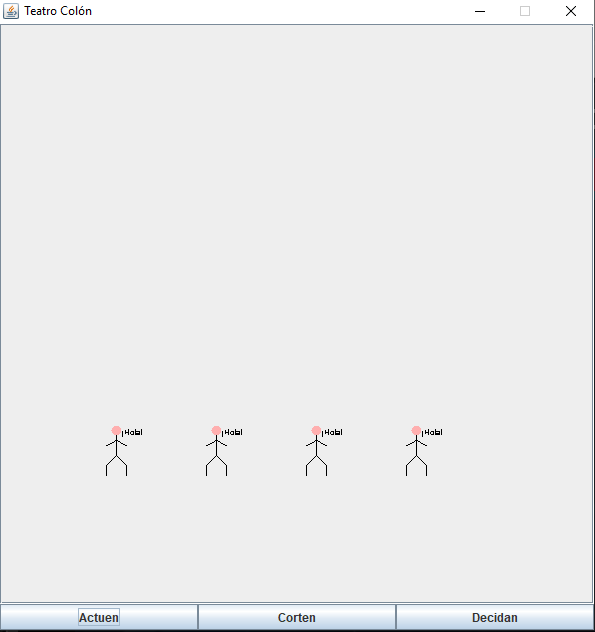


Después :

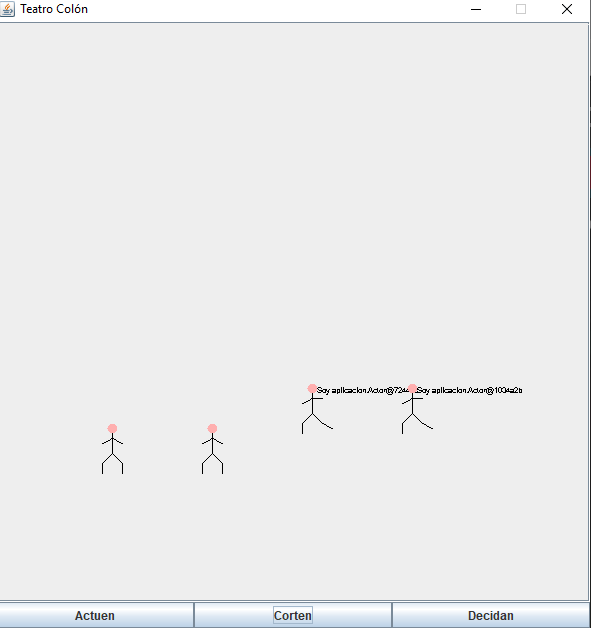


En corte:

Antes:

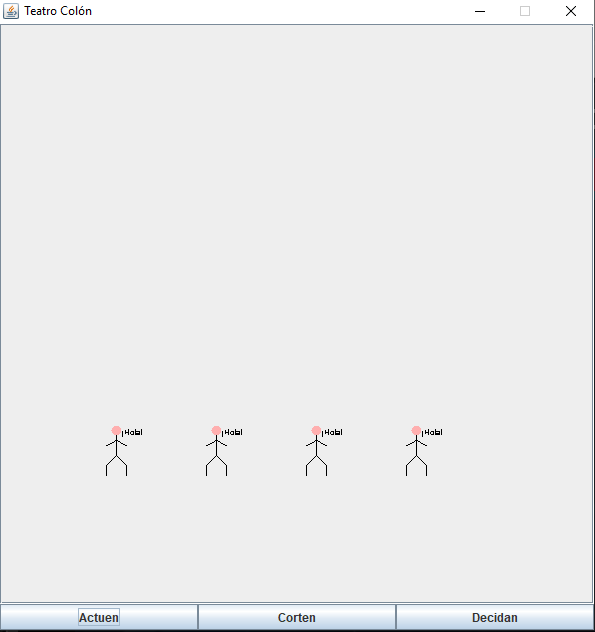


Después:

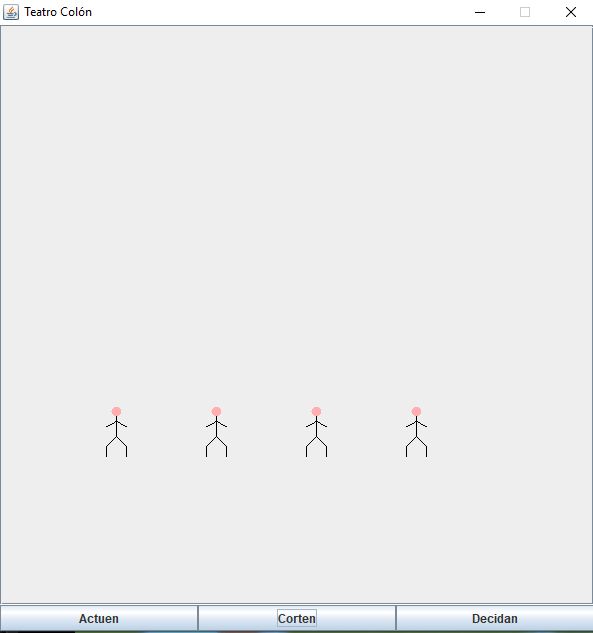


1. Los 4 actores quedan en corte es decir tanto con los brazos y piernas abajo

Antes:



Después:



**DE BLUEJ A CONSOLA En esta sección del laboratorio vamos a aprender a usar java desde consola. Para esto se va a trabajar con el proyecto del punto anterior.**

**Comandos básicos del sistema operativo [En lab03.doc]**

**Antes de iniciar debemos repasar los comandos básicos del manejo de la consola**

**1. Copiar un archivo**

Toca entrar hasta el directorio donde esta el archivo con CD y usar copy nombre\_archivo directorioNuevo

**Crear un archivo**

Toca entrar hasta el directorio donde esta el archivo con CD y usar COPY CON nombre\_archivo <ENTER> Texto que va a tener el archivo <ENTER> CTRL + Z

**Eliminar un archivo**

Toca entrar hasta el directorio donde está el archivo con CD y usar

DEL nombre\_archivo

**Listar un directorio**

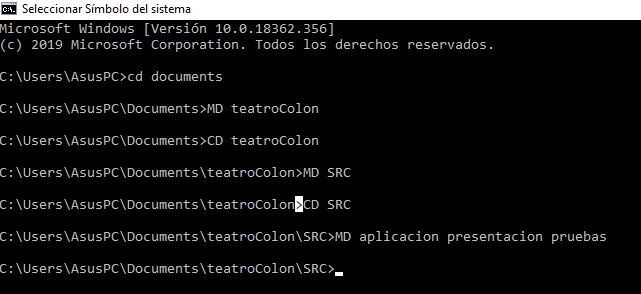
Colocas DIR directorio\_que\_quieres\_listar, y si lo quieres asignar a un archivo es DIR directorio\_que\_quieres\_listar > archivo al cual quieres asignarle el listado

**Crear un directorio**

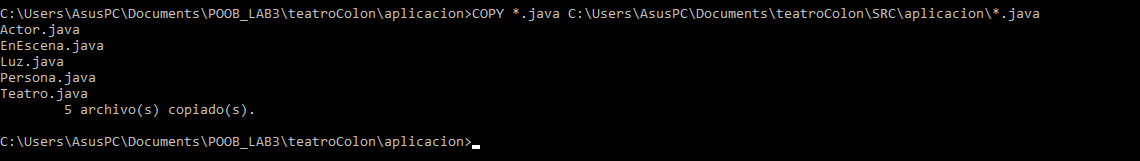
Toca entrar hasta el directorio que quieres agregar el subdirectorio y usar MD nombre\_del\_directorio

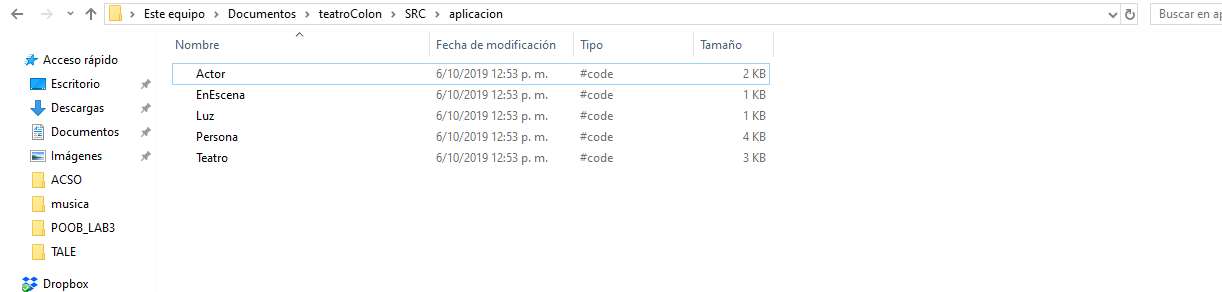
**Borrar un directorio**

Toca entrar hasta el directorio que quieres eliminar y usar RD nombre\_del\_directorio, si el directorio esta lleno usar RD nombre\_directorio /S

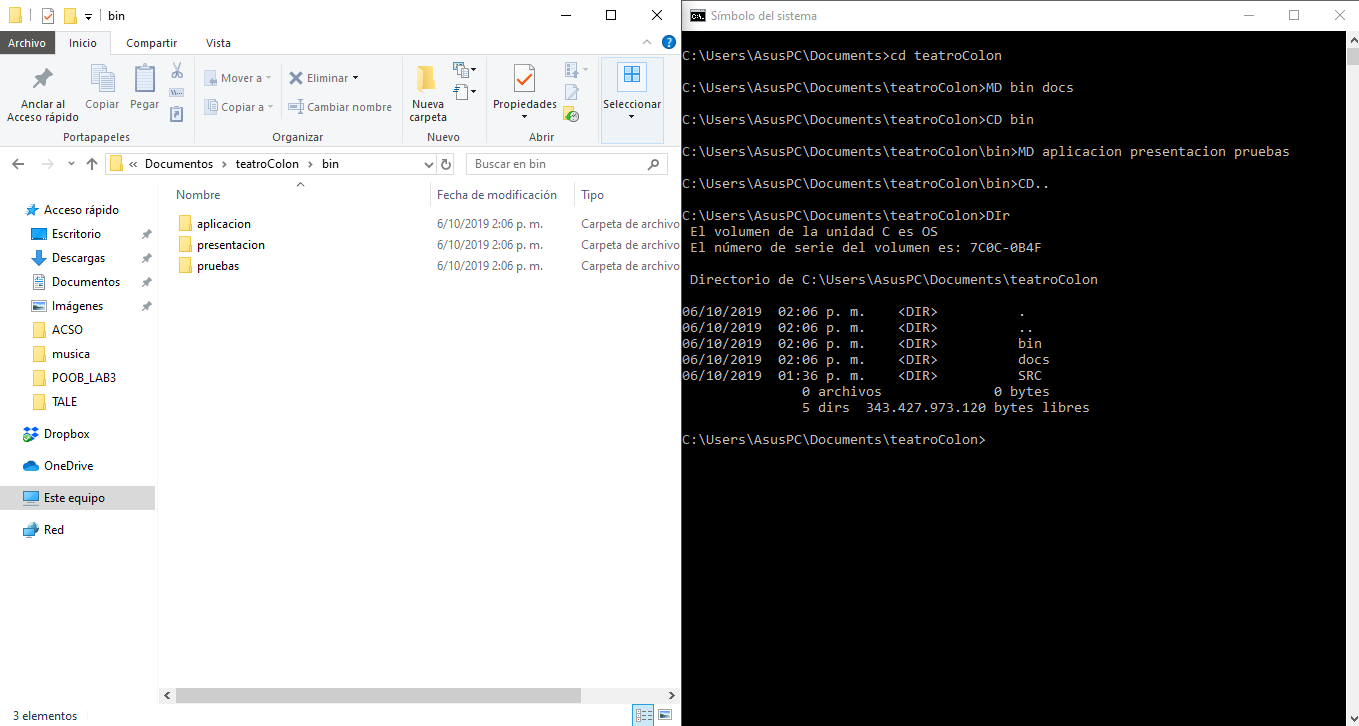
**2.**

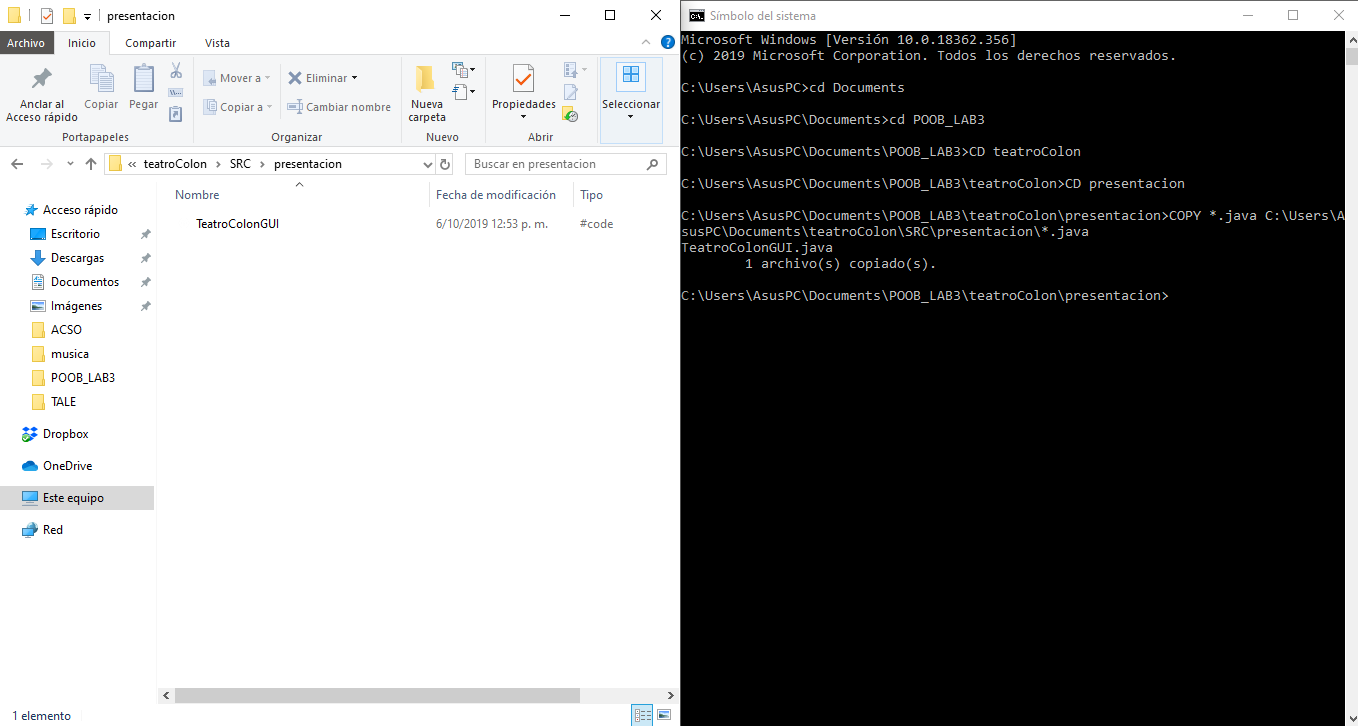
**3.**

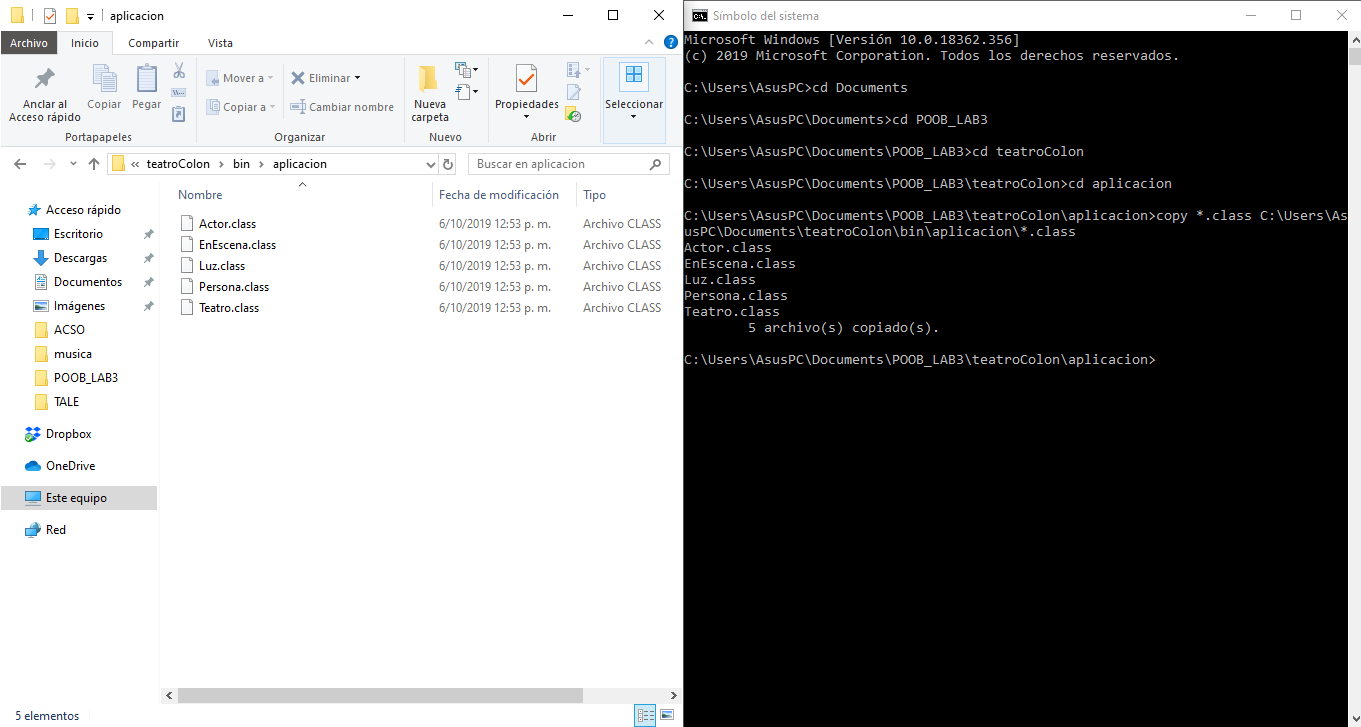
****

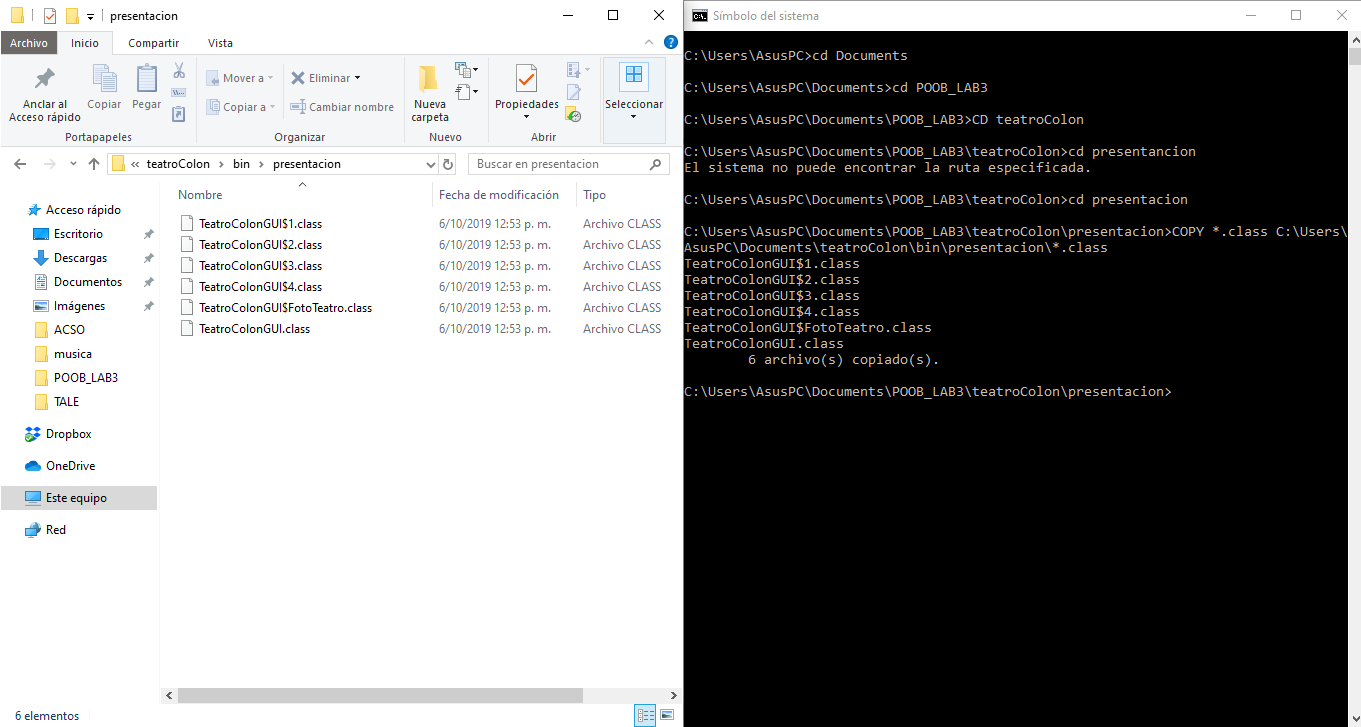
****

**Estructura de Proyectos Java**

**1.**

**2.**

****

****

**Comandos en Java**

**1.**

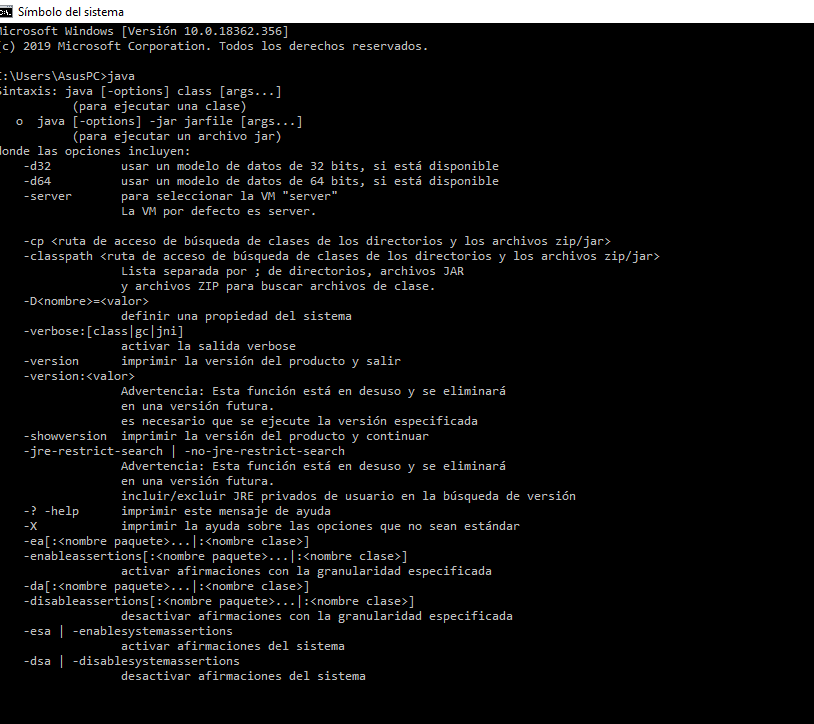
Javac es para compilar un programa Java desde el CMD}

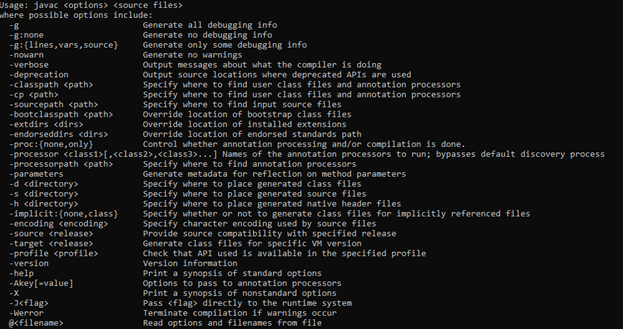
Java es para poder hacer varias cosas así como descomprimir

Jar es para comprimir un archivo

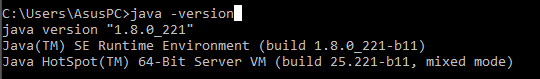
Javadoc es para generar la documentación de un archivo

**2.**

**JAVA**

**JAVAC**

**3.**



**Compilando.**

**1.**

Toca entrar hasta el directorio raíz del proyecto y usar el comando

javac -d bin src\aplicacion\\*.java src\presentacion\\*.java

**2.** No se generaron nuevos archivos en el directorio del Proyecto

**Documentando.**

**1.**Toca entrar hasta el directorio raiz y usar el commando

javadoc -d docs src\aplicacion\\*.java src\presentacion\\*.java

2. **.**El archivo que abrimos para poder empezar a navegar la documentación es Index

**Ejectuando.**

1.Toca entrar hasta el directorio raíz del proyecto y usar el comando

java -cp bin presentacion.TeatroColonGUI