Esteban Camilo Archila Bastidas

Sergio Andres Otero Herrera

Herencia e interfaces

**Conociendo**

1. En el directorio descarguen los archivos contenidos en artico.zip. Revisen el código de la aplicación

a) ¿Cuántos paquetes tiene?

2

b) ¿Cuántas clases tiene?

6

c) ¿Cuál es el propósito del paquete de presentación?

definir las acciones que puede realizar el ártico

d) ¿Cuál es el propósito del paquete de dominio?

definir de qué forma va a realizar esas acciones el ártico

2. En este laboratorio vamos a ejecutar la aplicación, no a solicitar servicios a objetos.

¿Qué método se usa para ejecutar una aplicación java?

se usa el metodo main

¿Qué clase tiene ese método?

ArticoGUI

3. Ejecuten el programa.

¿Qué funcionalidades ofrece?

accion, improvisen, cortes, camara rapida

¿Qué hace actualmente? ¿Por qué?

actualmente no realiza ninguna funcion, por que el constructor de artico es privado

**Arquitectura general**.

1. Consulten el significado de las palabras package e import de java.

el package hace referencia a un paquete y el import es la forma como podemos usar este paquete

¿Qué es un paquete?

La unidad donde se agrupan componentes relacionados.

¿Para qué sirve?

Para modular el código agrupando clases relacionadas.

Explique su uso en este programa.

Nos ayuda a separar la parte de definición, la parte de construcción dentro del código y para delegar correctamente las funcionalidades

2. Revisen el contenido del directorio de trabajo y sus subdirectorios. Describa su contenido.

tenemos 2 folders representando cada uno de los paquetes( dominio y presentacion) dentro de cada uno hay archivos .class .ctxt y .java

¿Qué coincidencia hay entre paquetes y directorios?

que los 2 funcionan como contenedores

3. Inicien el diseño con un diagrama de paquetes en el que se presente los componentes y las relaciones entre ellos.

En astah, crear un diagrama de clases (cambiar el nombre por Package Diagram0)

4. Ahora que conocen los paquetes, ¿cuál es la visibilidad real del modificador protected?

los métodos declarados del paquete pueden ser utilizados dentro de otras subclases o en otros paquetes

**Arquitectura detallada**

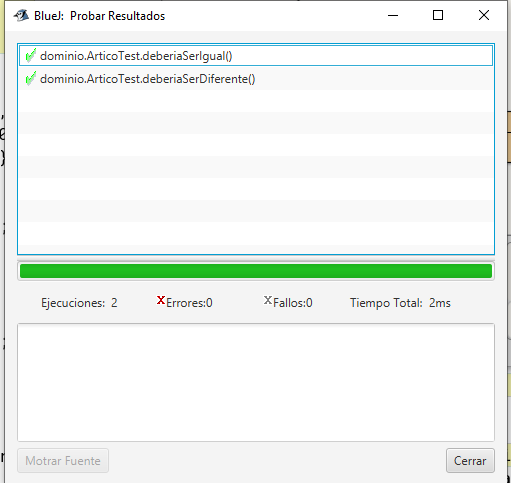
1. Usando ingeniería reversa prepararen el proyecto para MDD. Organicen el diseño estructural actual de la aplicación: realicen los diagramas de clases de los paquetes de presentación y dominio. Muevan las clases a los paquetes correspondientes.

(astah)

2. Adicionen en las fuentes la clase de pruebas necesaria para BDD. (No lo Adicionen al diagrama de clases) ¿En qué paquete debe estar? ¿Por qué? ¿Asociado a qué clase? ¿Por qué?

debe estar en el paquete de dominio por que este paquete contiene los metodos principales; se asocia con Artico por que

3. Escriban dos pruebas y Ejecutenlas. No olvide el estandar de nombres deberia y noDeberia. Presente un pantallazo con el resultado de las pruebas.



4. Estudie la clase Artico. ¿Qué tipo de colección se usa para albergar los elementos? ¿Puede recibir esquimales? ¿Por qué?

está usando ArrayList si puede recibirlos porque esquimales hereda de EnArtico

5. Estudie la clase Persona;

¿qué atributos pueden usar otras clases?

Arriba, Frente, Abajo

¿que atributos pueden usar sus subclases?

Arriba, Frente, Abajo, nombre, color

¿qué métodos no pueden cambiar las personas?

todos aquellos metodos con el modificador final

6. Estudie el código de la clase EnArtico;

¿qué atributos pueden usar otras clases?

Formas, r por ser default

¿qué métodos deben implementar las clases que están en Artico?

corte y accion

7. Revisen el contexto de los esquimales.

¿qué atributos son parte de su estado?

palabras

¿que atributos puede acceder en su código?

artico y palabras

¿qué métodos no pueden cambiar los esquimales?

accion

¿qué métodos pueden cambar parcialmente los esquimales?

el metodo mensaje por que usa el atributo protegido

8. Revisen el comportamiento de los esquimales:

¿Qué métodos deben tener el mismo comportamiento para todos los esquimales?

corte y accion

¿Qué métodos pueden cambiar totalmente?

corte y actue

¿Qué métodos pueden cambiar en parte? Explique.

mensaje

9. Completen el diagrama de clases de la capa de dominio

(astah)

**Ciclo 1. Esquimales normales**

1. Estudie el código de la clase Esquimal :

¿de qué color es?

negro, lo podemos ver en la clase super que en este caso seria “persona” y el atributo persona en el constructor esta seteado a negro

¿qué mensaje tiene?

la primera parte la podemos ver si nos vamos a la clase “persona” y el metodo mensaje

soy “nombre” :

para la segunda parte,si los parametros ingresados en “persona” de posicionx y posiciony, son inguales con la posicionx y posiciony de “Artico”

¡ENCONTRE EL POLO NORTE!

si no

¡Escalando!

¿qué hacen cuando dan orden de acción?

llama al metodo actue que llama al metodo de muevase y avance de persona

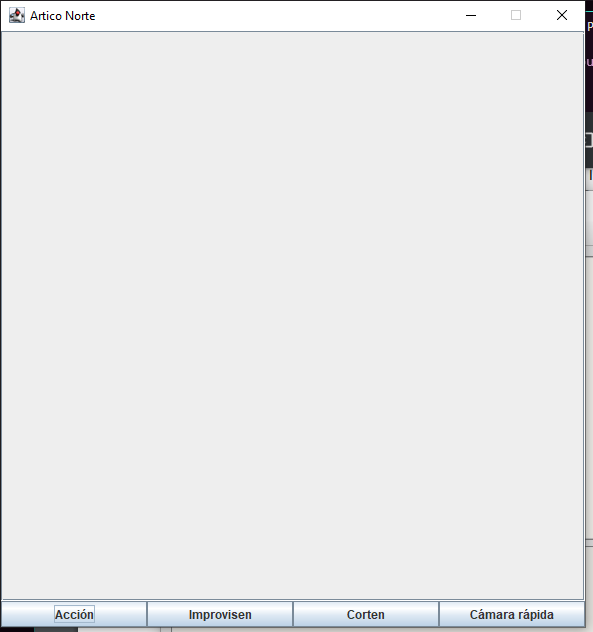
¿cuando cortan?

llama a los metodos muevaBrazo y muevaPierna de “Persona”

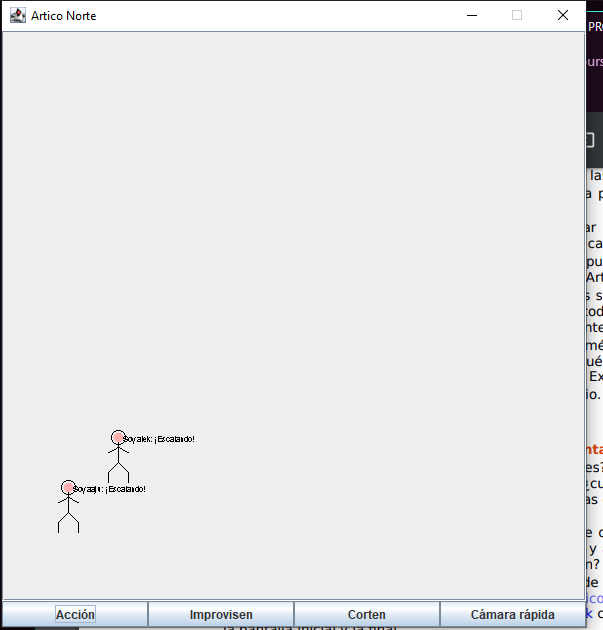
¿cuándo improvisan?

llama al metodo accion en “esquimal” y este llama a actue que luego llama a muevase y avance de “Persona”

2. En el método algunosEnArtico de la clase Artico cree dos esquimales en diferentes posiciones y acondiciónelos al Artico llámelos aaju y alek. Ejecuten el programa y Capturen la pantalla.



¿Qué pasa ahora?

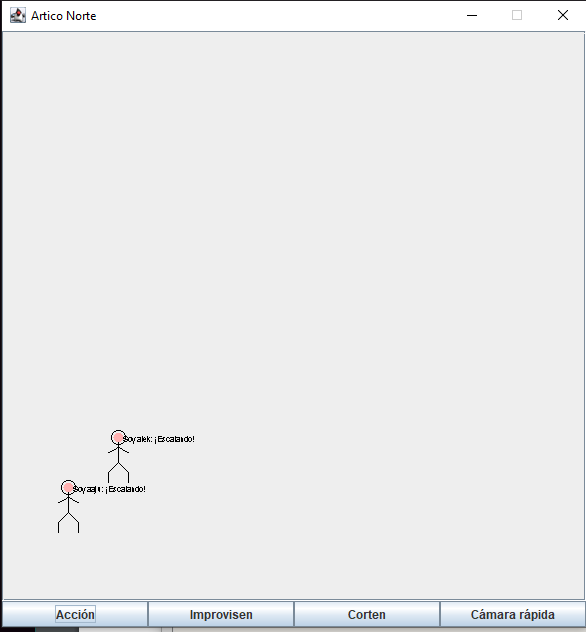


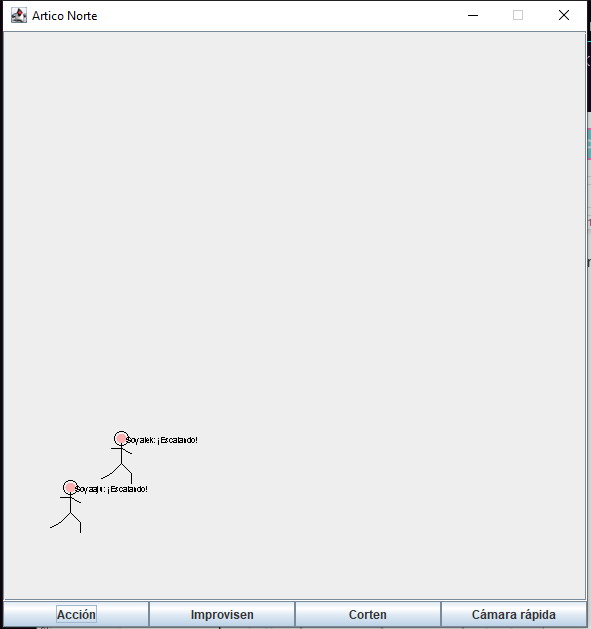
¿Pidales acción? ¿Qué pasa? ¿Por qué?

No pasa nada, el codigo de acción no esta creado en Artico.

3. En este punto vamos a construir el método que atiende el click del botón Acción de la interfaz: el método llamado acción() de la clase Artico. Ejecuten el programa y haga tres click en el botón Acción.

¿Actúan aaju y alek como lo esperaban? Capturen la pantalla inicial y la final.





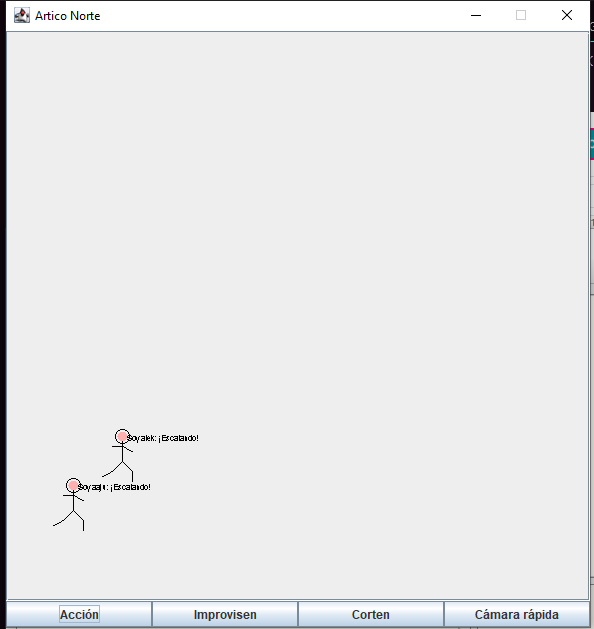
Si se comportan segun lo esperado

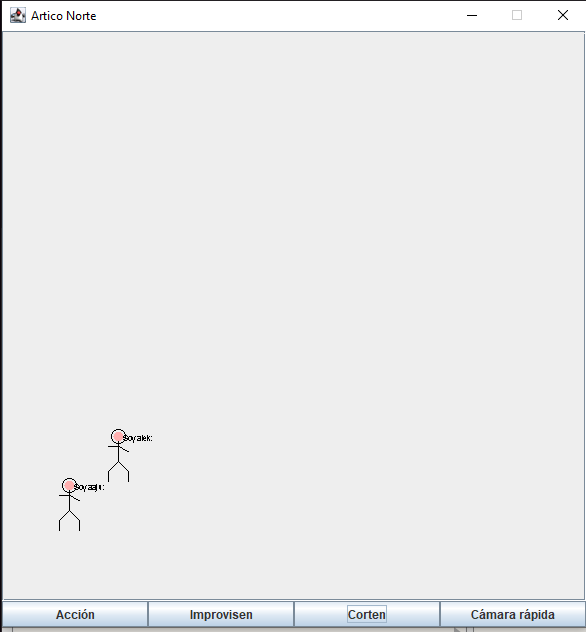
4. En este punto vamos a construir el método que atiende el click del botón Corten de la interfaz: el método llamado corten() de la clase Artico. Construya el método, Ejecuten el programa y haga click en el botón Corten.

¿Cómo quedan todos los esquimales después de esta orden?

Con los pies en el piso y sin palabras.

¿Es adecuado? Capturen la pantalla inicial y la final.





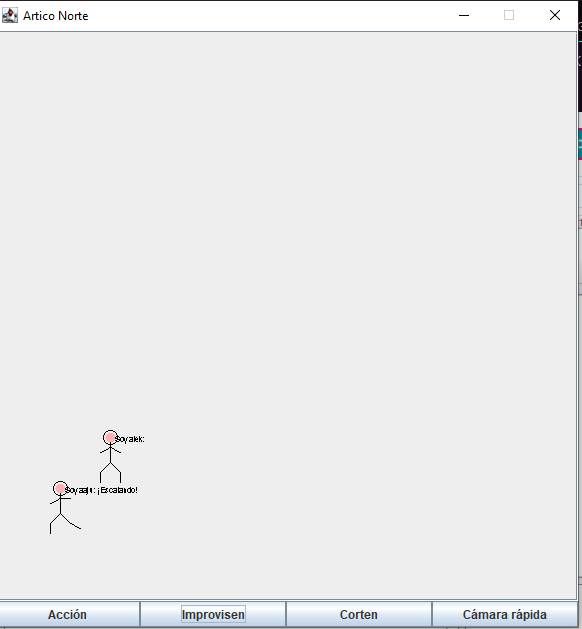
Si es adecuado, siguen lo especificado en el código de corten.

5. En este punto vamos a construir el método que atiende el click del botón Improvisen de la interfaz: el método llamado improvisen() de la clase Artico. Construya el método, Ejecuten el programa y haga click en el botón Improvisen.

¿Como quedan todos los esquimales después de esta orden?

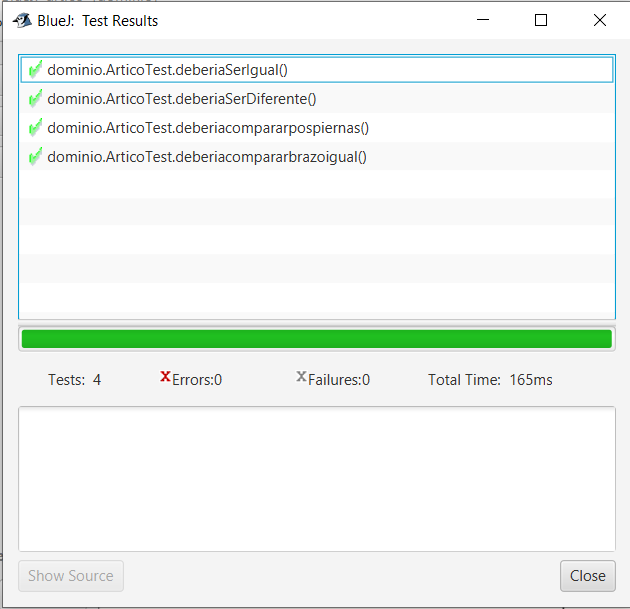
Algun(os) esquimales pueden realizar Corte() o Accion()

¿Es adecuado? Capturen la pantalla inicial y la final.



si es adecuado, hacen lo especificado en el codigo

6. Capturen el diseño de secuencia, el diagrama de clase y el resultado de ejecución de las pruebas de unidad. Expliquen.



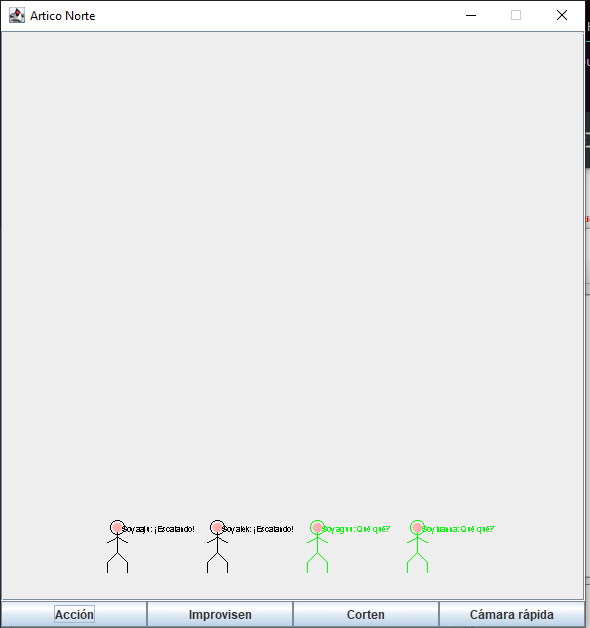
**Ciclo 2. Incluyendo a los esquimales sordos**

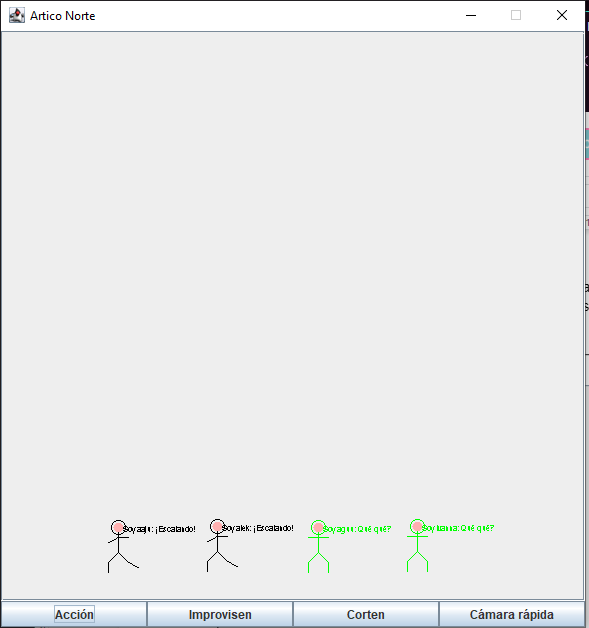
1. Implemente este nuevo esquimal.

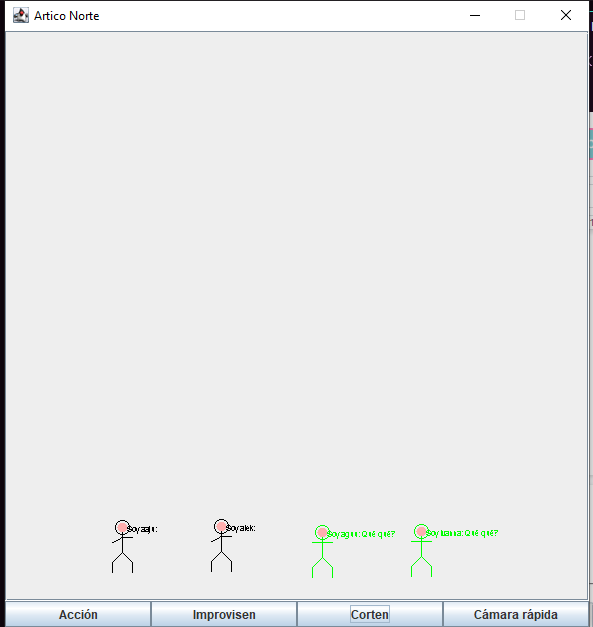
¿cuáles métodos se sobre-escriben (overriding)?

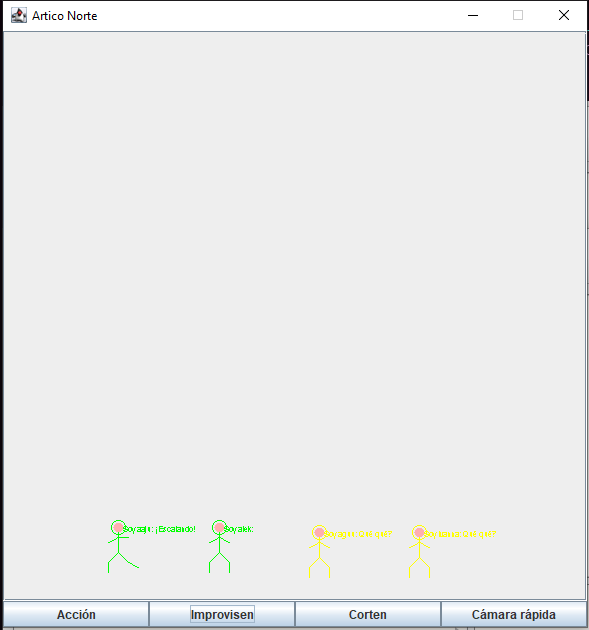
los metodos actue, improvisen, corte y palabras

2. Adicionen una pareja de esquimales sordos, llámelos aguu y ivanna, Ejecuten el programa y pídales a todos que actúen, corten e improvisen. Capturen una pantalla significativa. ¿Qué pasa?

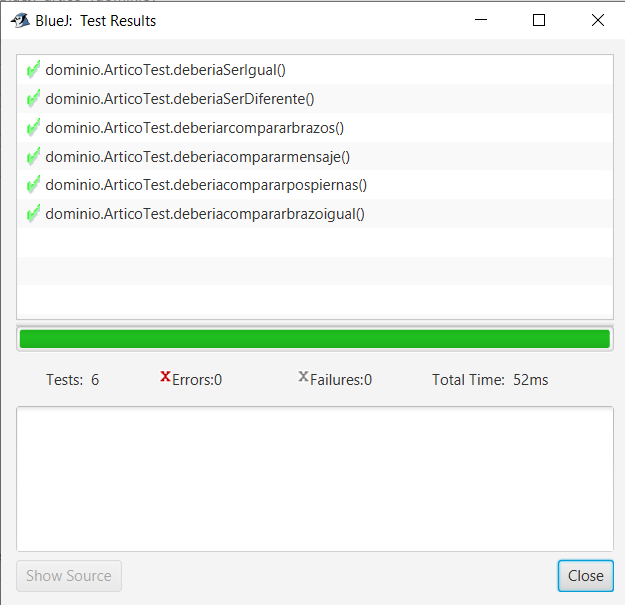








3. Presenten los cambios en el diseño de secuencia, los cambios en el diagrama de clase y el resultado de ejecución de las pruebas de unidad. Expliquen.



Si es correcto, cumple con las especificaciones del enunciad

**Ciclo 3. Adicionando iglus**

1. ¿Qué debemos hacer para tener iglús en la Artico?

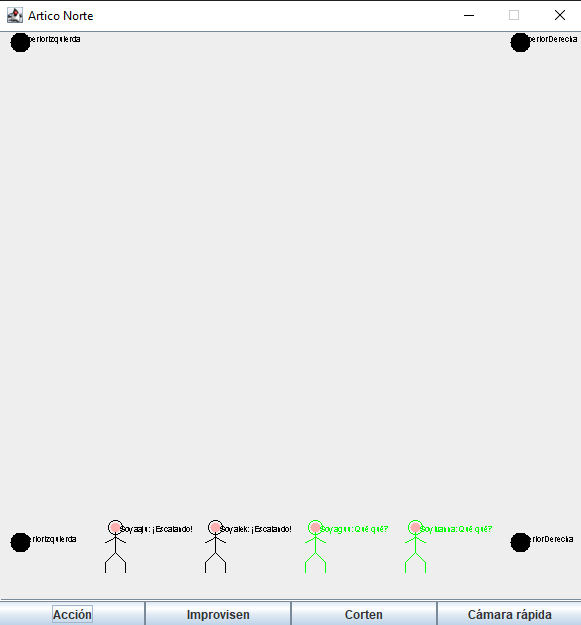
crear la clasi iglu y agregar cada iglu nuevo a la lista de elementos en la clase Artico

2. Construya la clase Iglu para poder adicionarla en el Artico. ¿qué cambios incluyó

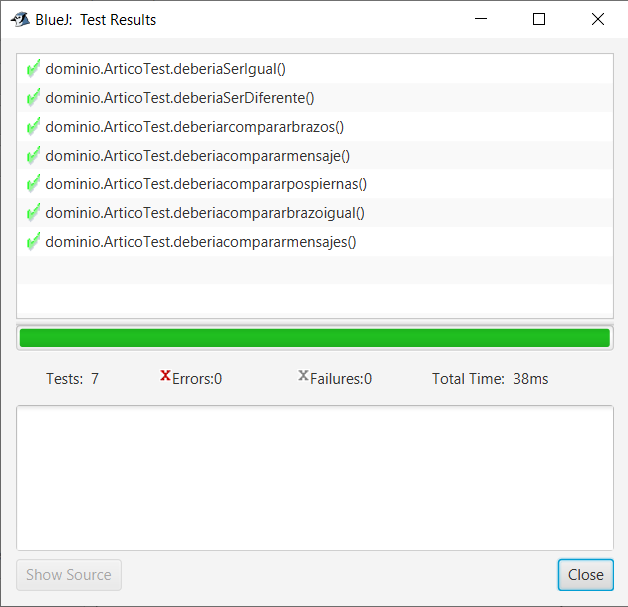
se construyeron los metodos para utilizar los botones de accion, improvisen y corten

3. Para aceptar este elemento , ¿debe cambiar en el código del Artico. en algo? ¿por qué?

no, simplemente se creo el nuevo objeto de tipo Iglu, por que para eso definimos todo dentro de la clase Iglu implementando a EnArtico

4. Adicionenn cuatro iglus en las esquinas del Artico, llámenlas superiorDerecha, superiorIzquierda, inferiorDerecha y inferiorIzquierda. Ejecuten el programa pídales a todos acción. Capturenn la pantalla. ¿Qué pasa? ¿es correcto?

5. Presenten los cambios en el diseño de secuencia, los cambios en el diagrama de clase y el resultado de ejecución de las pruebas de unidad. Expliquen.

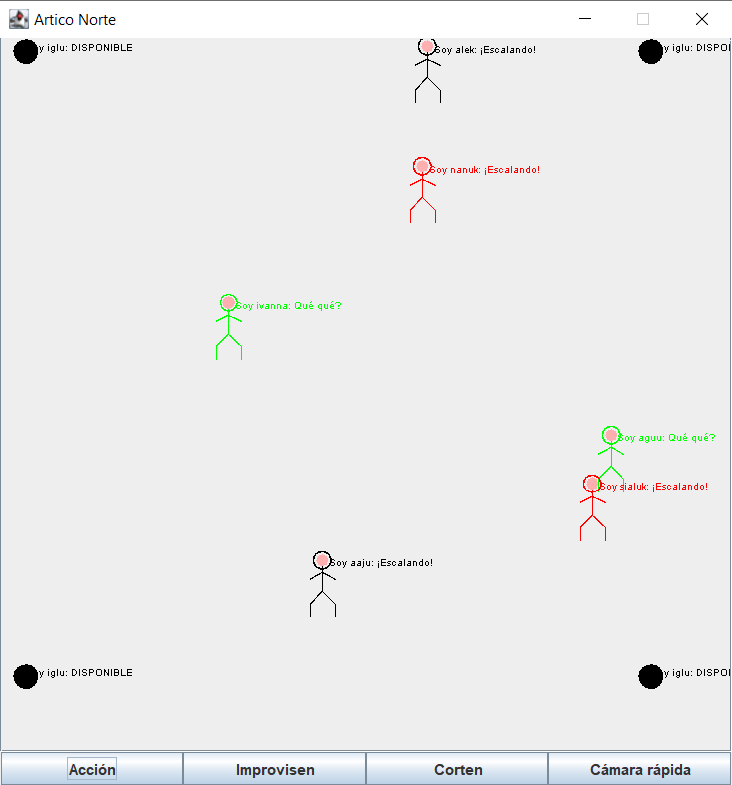


**Ciclo 4. Creando un nuevo esquimal: el explorador**

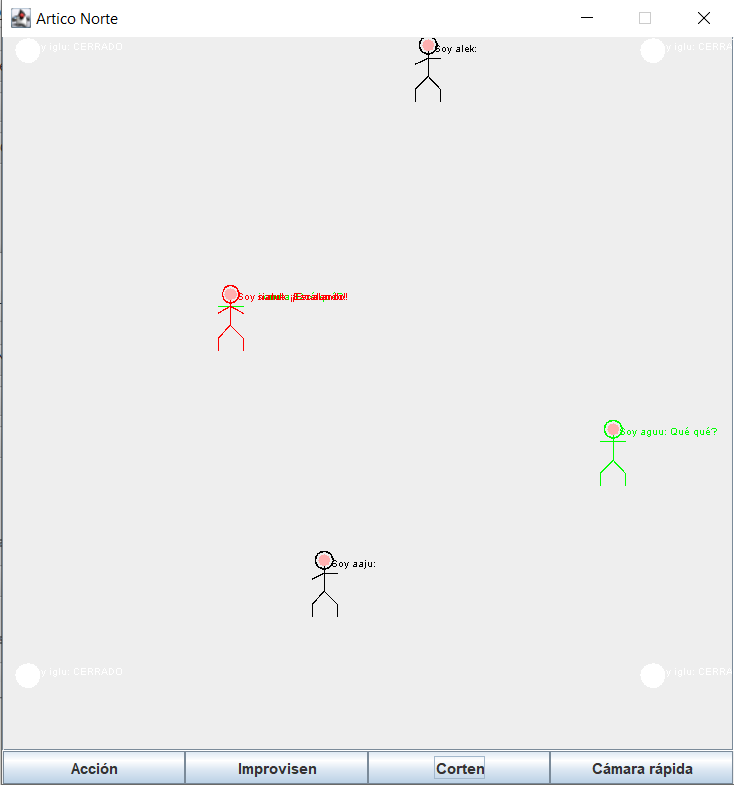
1. Implemente este nuevo esquimal. ¿cuáles métodos se sobre-escriben (overriding)?

los metodos actue, improvise y corte

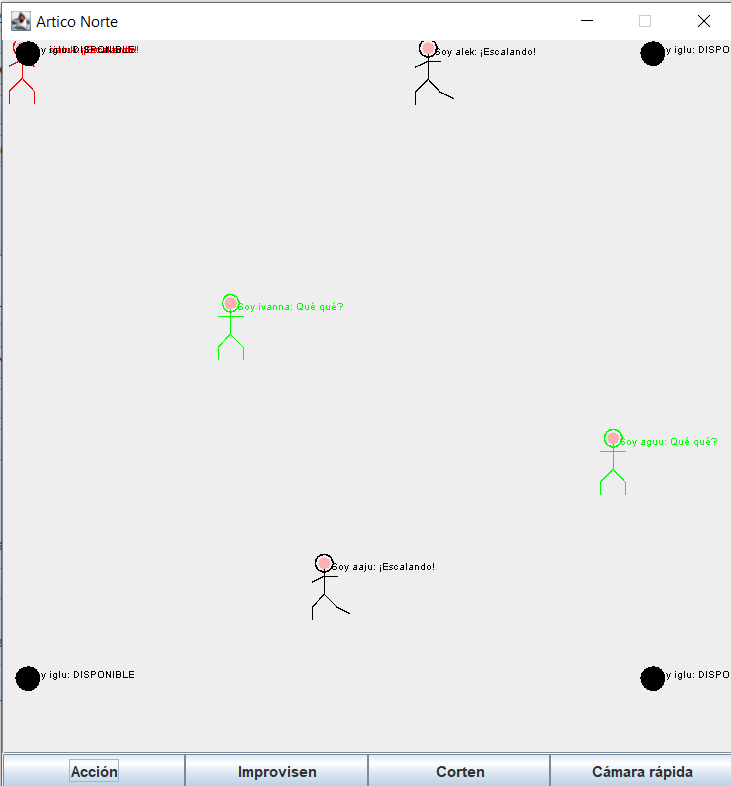
2. Adicionen una pareja de esquimales minuciosos, llámelos nanuk y sialuk, Ejecuten el programa y pídales a todos que actúen y que paren. Capturen la pantalla. ¿Qué pasa?



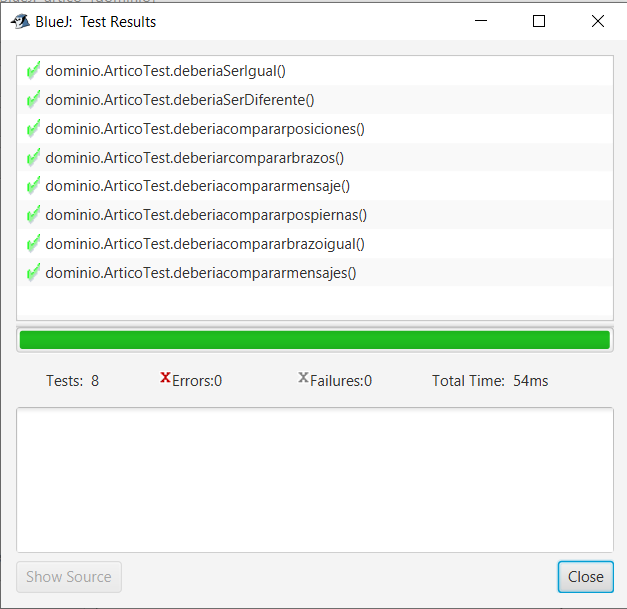
cuando cortan se acercan al Esquimal mas cercano



al pedirles que actuen los exploradores recorren todo el polo hasta llegar a el iglu superior derecho



3. Presenten los cambios en el diseño de secuencia, los cambios en el diagrama de clase y el resultado de ejecución de las pruebas de unidad. Expliquen.



**Ciclo 5. Nuevo esquimal: Proponiendo y diseñando**

1.Propongan, describan e implementen un nuevo tipo de esquimal.

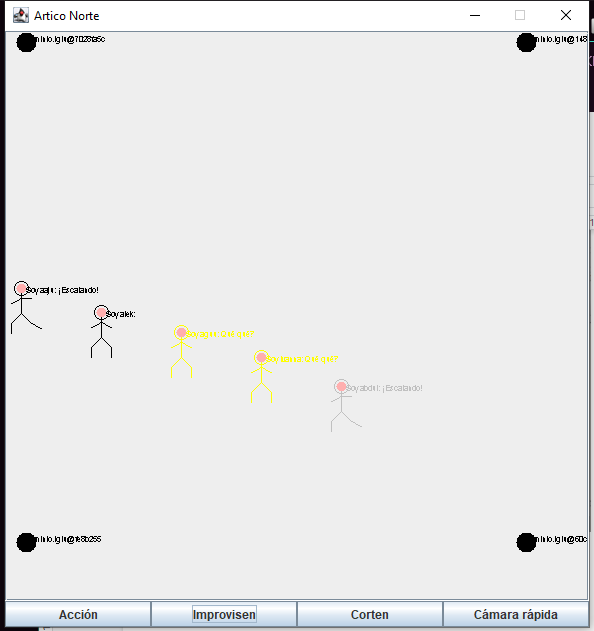
se creó el nuevo esquimal “EsquimalExperto”

2. Incluyan una pareja de ellos con el nombre de ustedes. Ejecuten el programa con dos casos significativos. Explique la intención de cada caso y capturen las pantallas correspondientes.

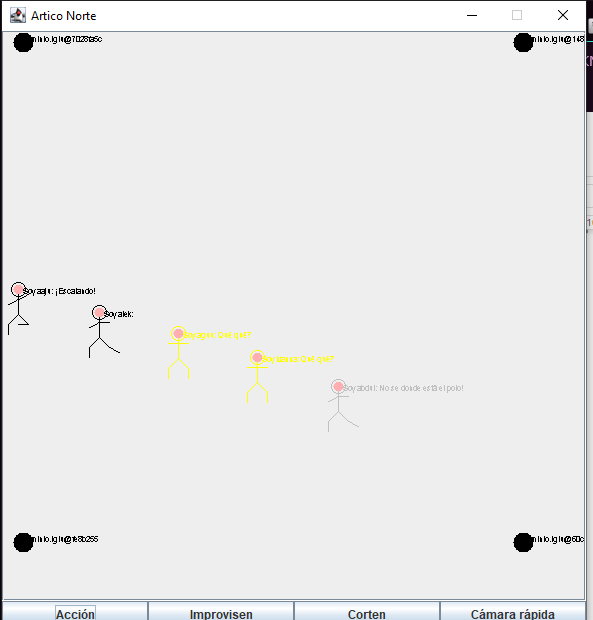


Se crea el nuevo tipo de esquimal: EsquimalExperto.

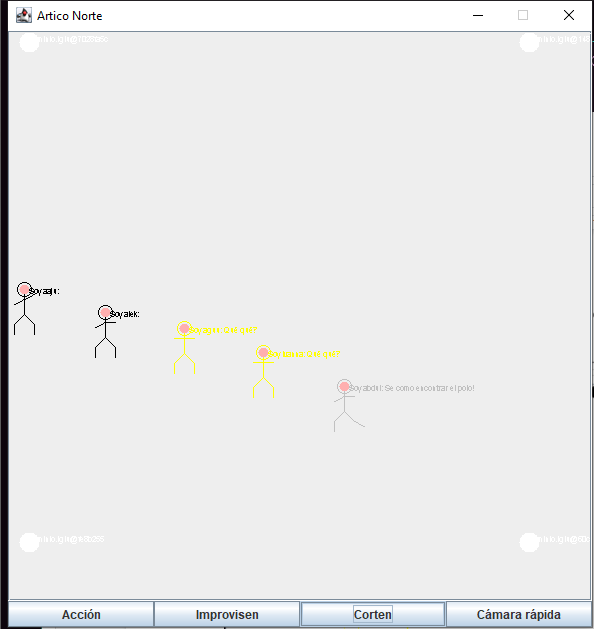
Si se hace Improvise(), mueve brazos y piernas



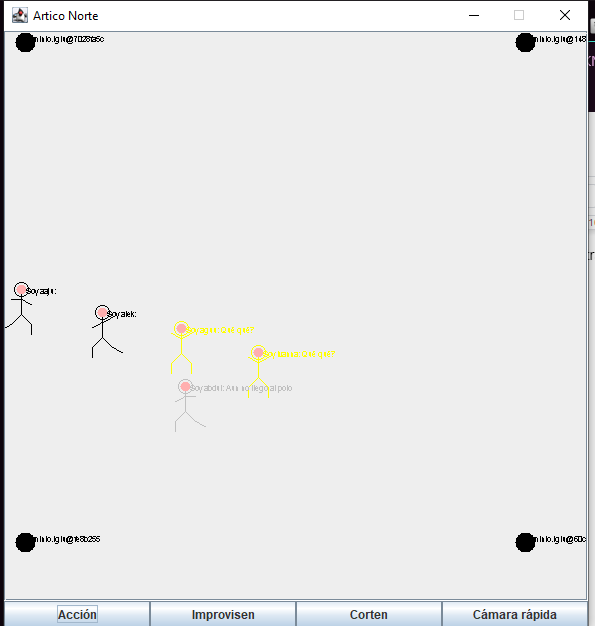
Si se hace Accion() antes de Corte(), cambia el mensaje del Esquimal a “No se donde esta el polo”



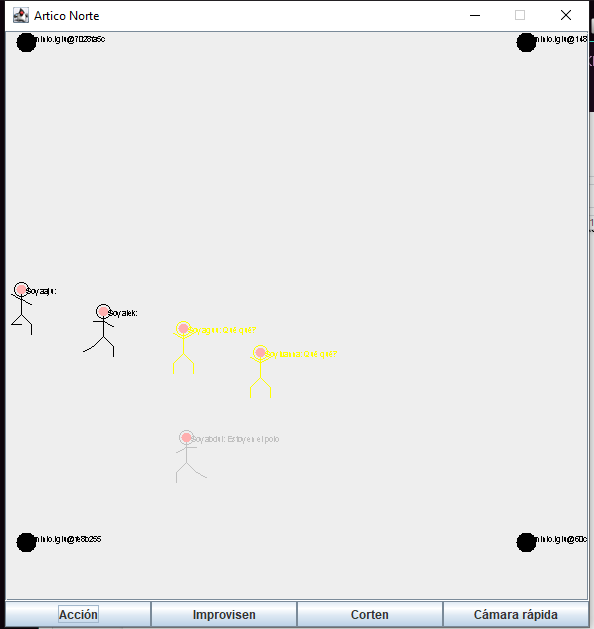
Si se hace Corte() cambia el mensaje a “Se como encontrar el polo!”



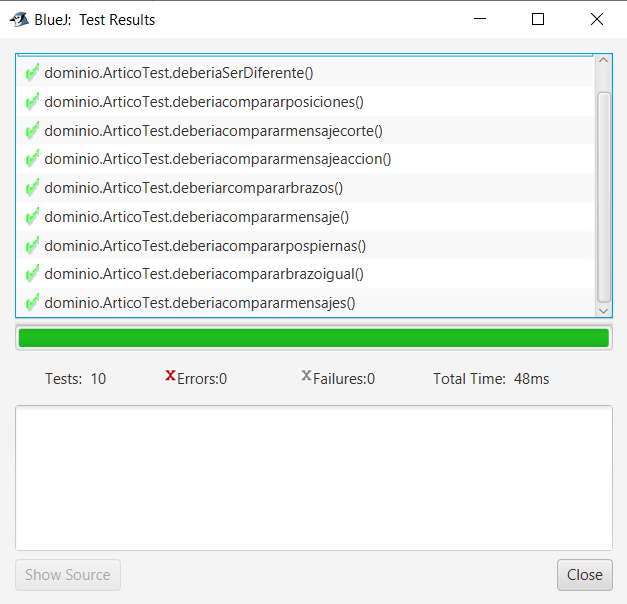
Luego de haber usado corte, como el EsquimalExperto ya conoce la posicion del polo, se desplaza primero en el eje X, cambiando el mensaje a “Aun no llego al polo!”



Por ultimo, cuando se usa Accion() de nuevo, se desplaza en el eje Y, indicando finalmente su llegada al polo con el mensaje “Estoy en el polo!”



3. Presenten los cambios en el diseño de secuencia, los cambios en el diagrama de clase y el resultado de ejecución de las pruebas de unidad. Expliquen.

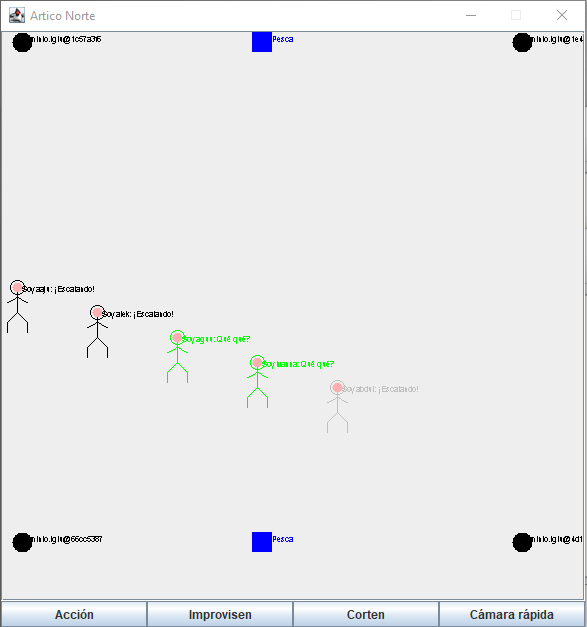


**Ciclo 6. Nuevo elemento: Proponiendo y diseñando**

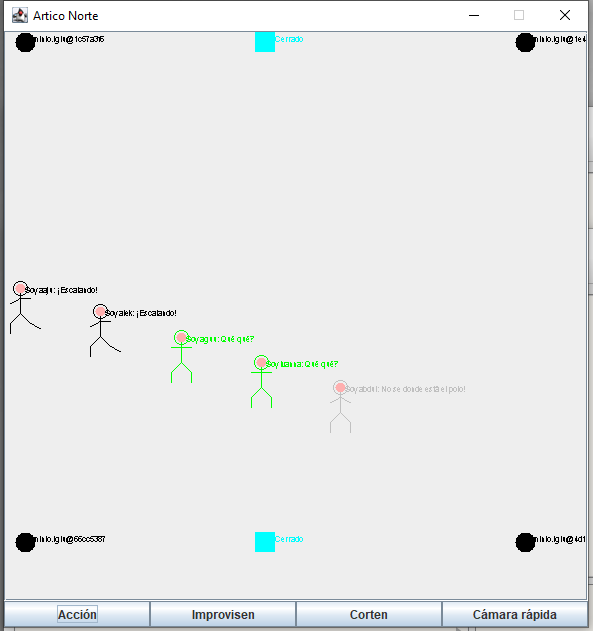
1.Propongan, describan e implementen un nuevo tipo de elemento

Se creó la nueva clase “HuecoPesca”

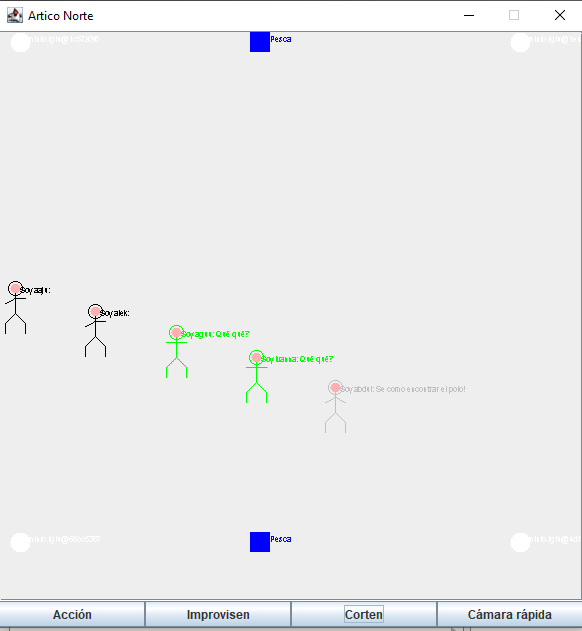
2. Incluyan dos de ellos con el nombres semánticos. Ejecuten el programa con dos casos significativos. Explique la intención de cada caso y Capturen las pantallas correspondientes.



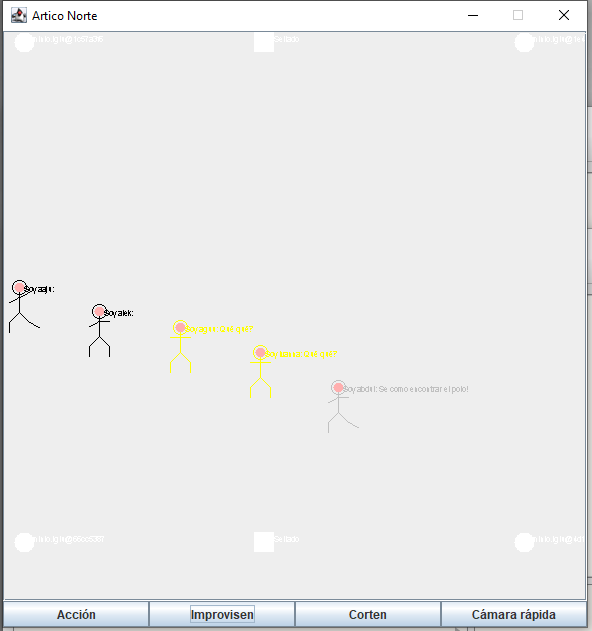
Inicialmente se crea un HuecoPesca color azul, disponible para pescar



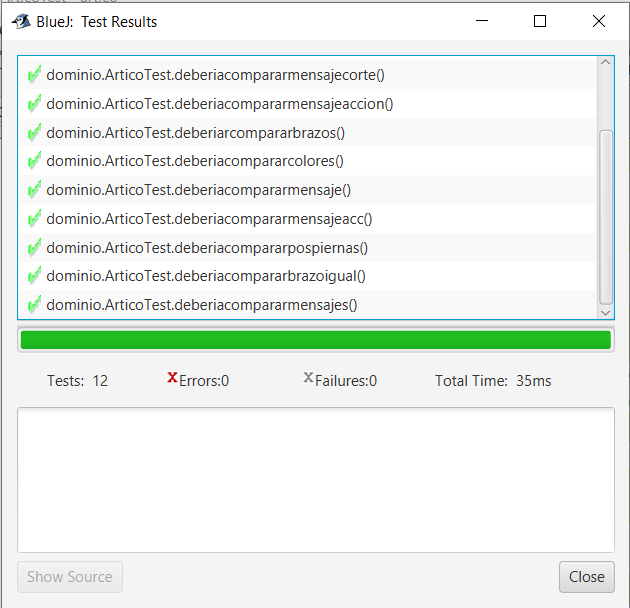
Cuando se usa Accion(), el HuecoPesca queda cerrado, solo se puede abrir con el boton de Corten(), como podemos ver a continuacion:



Por ultimo, si se da al boton Improvisen(), el HuecoPesca queda sellado y con color blanco, no hay forma de volverlo a abrir o cambiar este estado

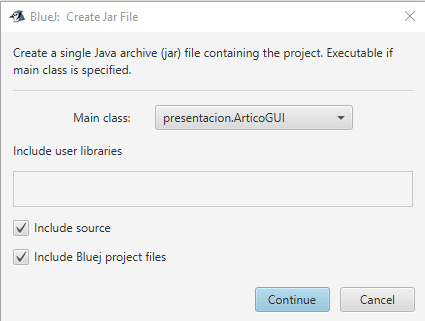


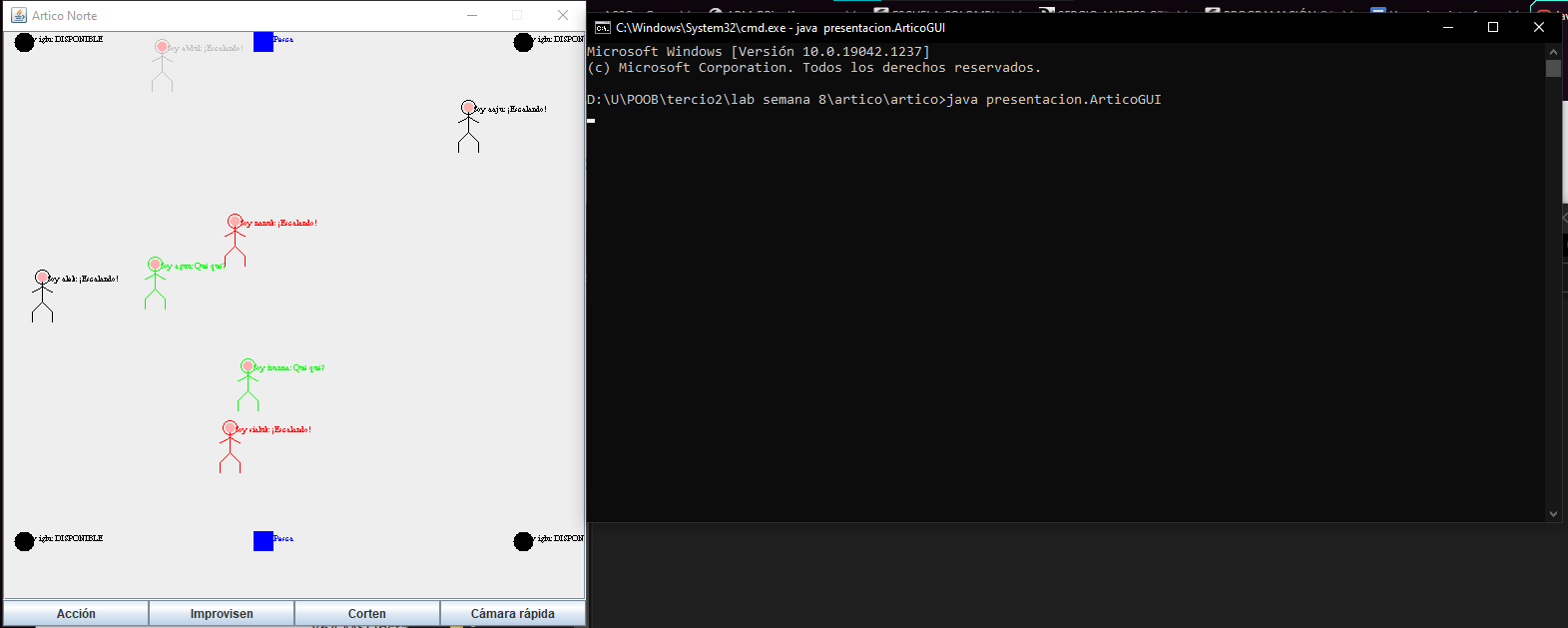
3. Presenten los cambios en el diseño de secuencia, los cambios en el diagrama de clase y el resultado de ejecución de las pruebas de unidad. Expliquen.



**Empaquetando la versión final para el usuario.**

1. Revisen las opciones de BlueJ para empaquetar su programa entregable en un archivo .jar. Genere el archivo correspondiente.





2. Consulten el comando java para ejecutar un archivo jar. Ejecutenlo ¿qué pasa?

“***java ‘-jar c: rutadelarchivo.jar***”.

Genera el artico sin necesidad de entrar al codigo.

3. ¿Qué ventajas tiene esta forma de entregar los proyectos? Explique claramente.

Cualquier persona puede abrir esto sin necesidad de entrar al codigo fuente, garantizando su integridad.

**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.

crear: **TYPE ARCHIVO.EXTENSION**

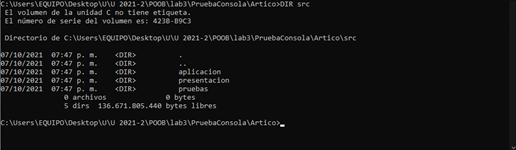
borrar: **DEL**

listar: **DIR**

copiar: **COPY ARCHIVO DESTINO**

eliminar: **DEL ARCHIVO O CARPETA**

2. Organicen un nuevo directorio con la estructura propuesta para probar desde allí su habilidad con los comandos de consola. Consultenn y Capturenn el contenido de su directorio Artico src aplicacion presentacion pruebas



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



**Estructura de proyectos java**

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

todos los archivos .class por que son los que el computador puede ejecutar



**Comandos de java**

1. Consulten para qué sirven cada uno de los siguientes comandos:

javac: compila el código fuente

java:ejecuta el código

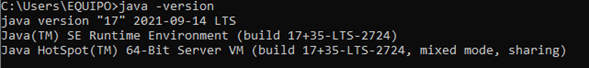
javadoc: genera la documentación en HTML

jar: ejecuta el archivp .jar

2. Cree una sesión de consola y Consulten en línea las opciones de los comandos java y javac. Capturen las pantallas.

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

la opción es java -version



**Compilando**

1. Utilizando el comando javac, desde el directorio raiz (desde artico con una sóla instrucción), compile el proyecto.

¿Qué instrucción completa tuvo que dar a la consola para compilar TODO el proyecto?



**Ejecutando**

1. Empleando el comando java, desde el directorio raiz, Ejecuten el programa. ¿Cómo utilizó este comando?



**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). ¿Cómo empaquetó jar ?



**RETROSPECTIVA**

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/ Hombre)

Camilo Archila = 23 horas

Sergio Otero = 23 horas

2. ¿Cuál es el estado actual de laboratorio? ¿Por qué? (Para cada método incluya su estado)

Casi completado, por que algunas pruebas de consola no funcionaron

3. Considerando las prácticas XP del laboratorio de hoy ¿por qué consideran que son importante?

por que asi el diseno y codificacion quedan mucho mas limpios y claros

4. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor problema? ¿Qué hicieron para resolverlo?

hacer correctamente los 6 ciclos por que se utilizaron muchas herramientas nuevas, el mayor problema fue correr el .jar desde la consola

5. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

complementar las dudas del otro, realizar mas pruebas