**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Herencia e interfaces**

**ADEMAS Java desde consola**

**2020-2**

**Laboratorio 3/6**

**Conociendo**

1. **En el directorio descarguen los archivos contenidos en islaTortuga.zip. Revisen el código de la aplicación**
2. **¿Cuántos paquetes tiene?**

Tiene 2 paquetes, presentación y dominio.

1. **¿Cuántas clases tiene en total?**

Tiene 7 clases.

**c) ¿Cuál es la clase con la que se ejecuta la aplicación? ¿Por qué?**

IslaGui, porque tiene el método principal(main()).

1. **Ejecuten el programa.**

**¿Qué funcionalidades ofrece?**

Actúen, Aprisa, Paren, Decidan.

**¿Qué hace actualmente?**

Actualmente no realiza ninguna función.

**¿Por qué? (Deben ejecutar la aplicación java, no crear un objeto como lo veníamos haciendo)**

Porque dentro de la clase Isla no se ha implementado ninguno de los métodos para que funcione.

**Arquitectura general.**

1. **Consulte el significado de las palabras package e import de java.**

**¿Qué es un paquete?**

Es una colección de clases que generalmente están relacionadas.

**¿Para qué sirve?**

Sirve para relacionar las partes de un programa que se puede organizar como una unidad, y sirve en el mecanismo de control de acceso de Java.

1. **Revise el contenido del directorio de trabajo y sus subdirectorios. Describa su contenido.**

**¿Qué coincidencia hay entre paquetes y directorios?**

La coincidencia es que cada paquete esta guardado también como un directorio.

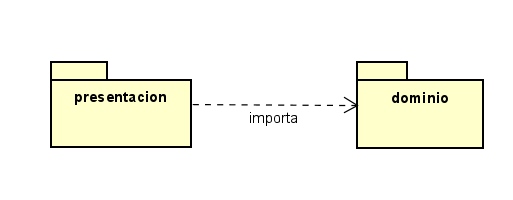
1. **Revisando las clases que contiene, ¿cuál es el propósito de cada uno de los paquetes encontrados?**

Presentación: El propósito de esta es todo lo relacionado co la interfaz, es mostrarnos la aplicación con sus botones.

Dominio: En el paquete dominio se encuentra toda la parte lógica del programa, porque se encuentran todo el desarrollo de los métodos que ayudan al funcionamiento de la aplicación.

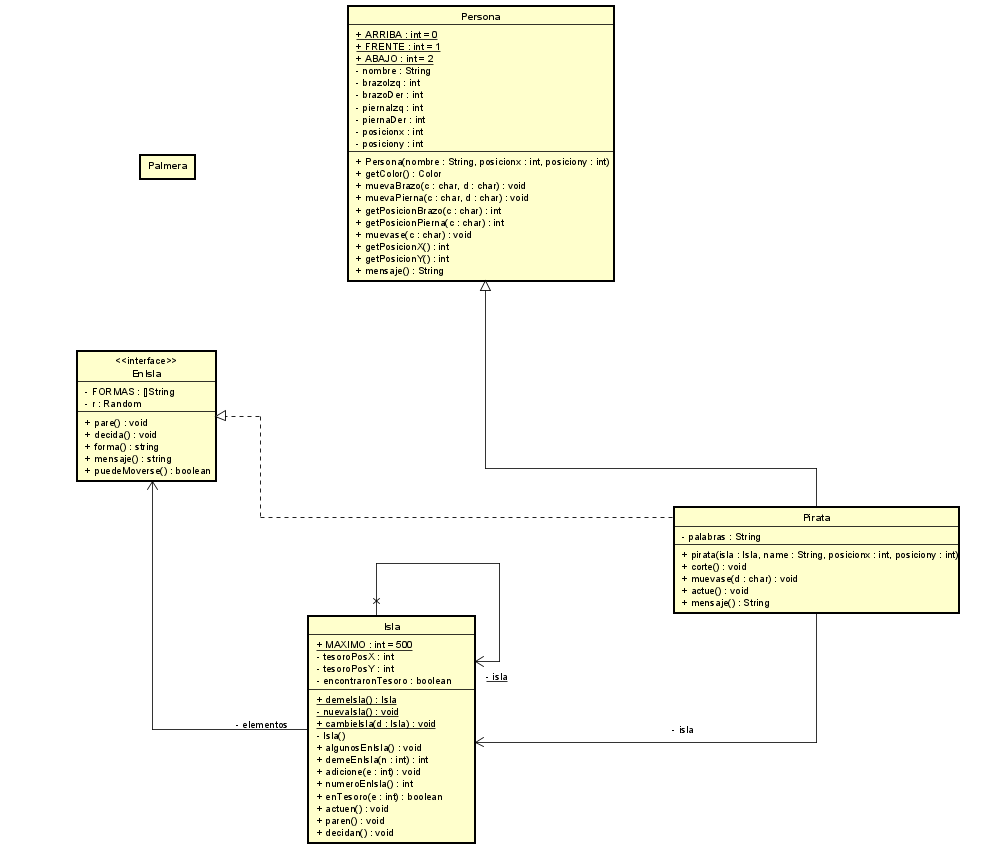
1. **Adicione el diseño de diagrama de paquetes en el que se presente los componentes y las relaciones entre ellos.**

**En astah, crear un diagrama de clases (cambiar el nombre por Diagrama de Paquetes)**

****

**Arquitectura detallada.**

1. **Usando ingeniería reversa prepararen el proyecto para MDD. Completen el diagrama de clases del paquete de dominio. No incluyan el paquete de presentación.**

****

1. **Adicione en las fuentes la clase de pruebas necesaria para BDD. (No lo adicione al diagrama de clases)**

**¿En qué paquete debe estar?**

En el paquete de Dominio

**¿Por qué?**

Porque en ese paquete se encuentra toda la lógica del programa

**¿Asociado a qué clase?**

En la clase Isla

**¿Por qué?**

Porque es donde se implementan las funciones principales

1. **Estudie la clase Isla.**

**¿Qué tipo de colección se usa para albergar los elementos?**

ArrayList con elementos de tipo EnIsla

**¿Puede recibir piratas? ¿Por qué?**

Si, porque pirata implementa el comportamiento de la interfaz EnIsla

1. **Estudie la clase Persona;**

**¿qué atributos pueden usar otras clases?**

Color, ARRIBA, FRENTE, ABAJO.

**¿qué atributos pueden usar sus subclases?**

Color.

**¿qué métodos no pueden cambiar las personas?**

muevaBrazo, muevaPierna, getPosicionBrazo, getPosicionPierna, getPosicionX, getPosicionY.

1. **Estudie el código de la clase Pirata;**

**¿qué atributos pueden usar otras clases?**

Palabras

**¿qué atributos pueden usar sus subclases?**

palabras

**¿qué métodos no pueden cambiar los piratas?**

Muevase, mensaje

**¿qué métodos pueden cambiar parcialmente los piratas?**

Ninguno

1. **Estudie el código de la clase EnIsla;**

**¿qué atributos pueden usar otras clases?**

Pueden usar todos los atributos.

**¿qué métodos deben implementar las clases que están enisla?**

Ninguno porque todos los métodos que hay enisla tienen cuerpo y están definidos como default.

**Ciclo 1. Actúan y descansan los Piratas NO OLVIDE BDD – MDD) (Construir: diseñar, implementar y probar)**

1. **Estudie el código de la clase Pirata:**

**¿de qué color es?**

El color es negro, es un atributo de la clase

**¿qué mensaje tiene?**

Encontré el tesoro, en el método mensaje

**¿cómo actúan?**

Mira la posición en la que tienen sus extremidades y se mueve

**¿qué hacen cuando paran?**

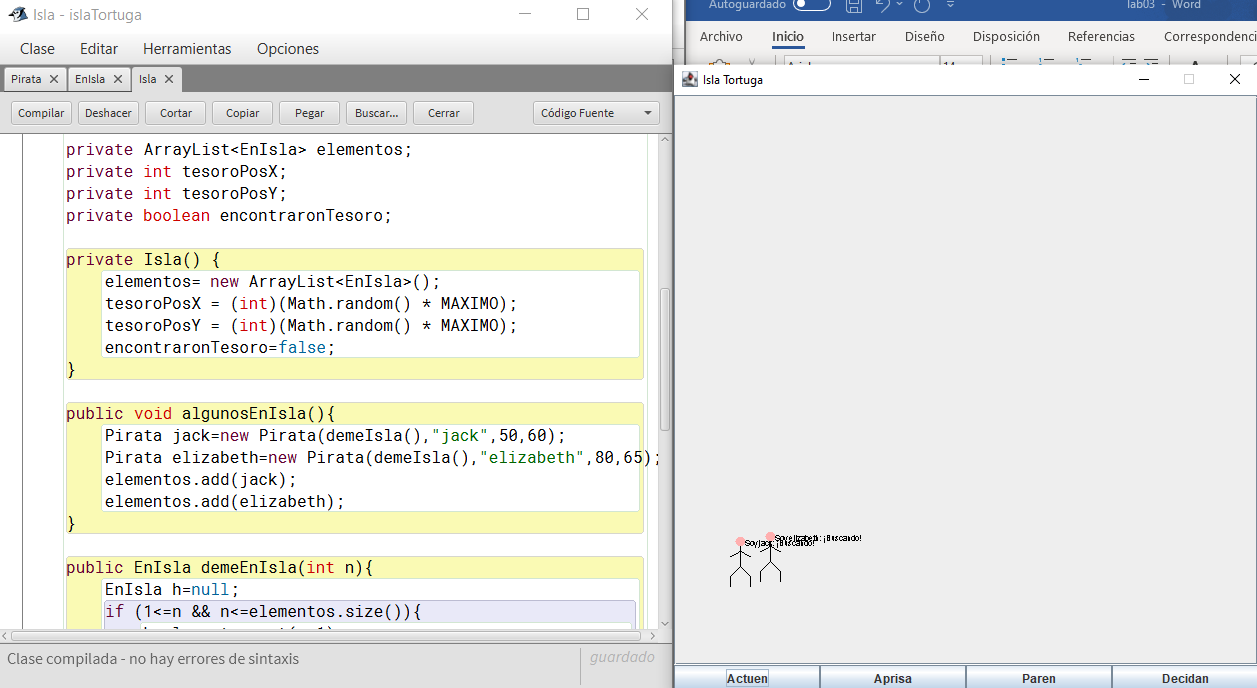
Han encontrado el tesoro, en el método muévase verifica si se a encontrado el tesoro

**¿cómo deciden?**

Aleatoriamente, en el método actue () hay un random

**Explique claramente dónde está la información de las respuestas de cada una de las preguntas.**

1. **En el método algunosEnIsla de la clase Isla cree dos piratas en diferentes posiciones y acondiciónelos al isla llámelos jack y elizabeth. Ejecute el programa y capture la pantalla.**

****

**¿Qué pasa ahora?**

Se muestran dos piratas con sus nombres.

**Pídales que actúen**

**¿Qué pasa?**

No implementa la funcionalidad

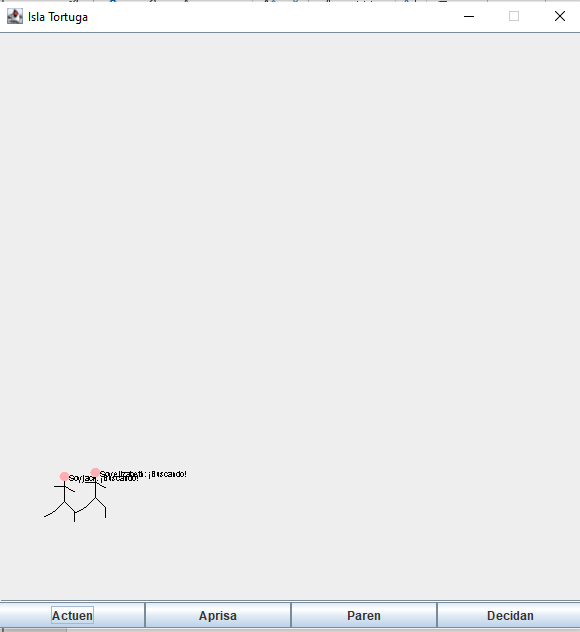
**¿Por qué?**

En la clase Isla no se ha implementado el método actúen.

1. **En este punto vamos a construir el método que atiende el click del botón Actúen de la interfaz: el método llamado actuen() de la clase Isla. Ejecute el programa y haga tres click en el botón Actuen.**

**¿Cómo actúan jack y elizabeth? Capture la pantalla inicial y la final.**

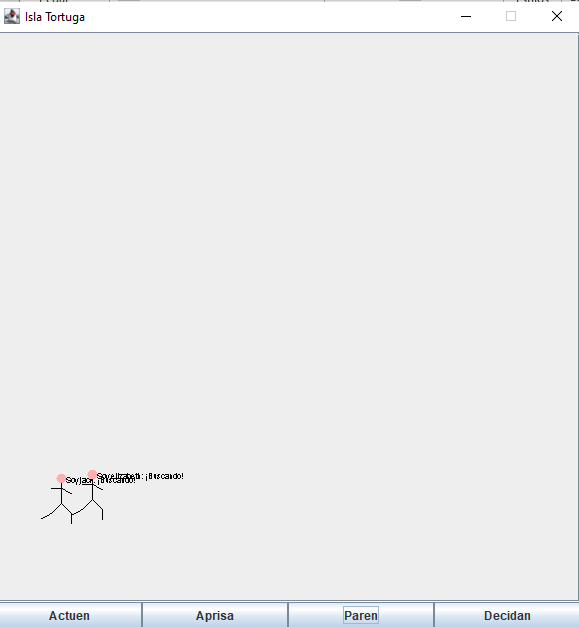
Se mueven sus extremidades

****

1. **En este punto vamos a construir el método que atiende el click del botón Paren de la interfaz: el método llamado paren() de la clase Isla. Construya el método, ejecute el programa y haga click en el botón Paren.**

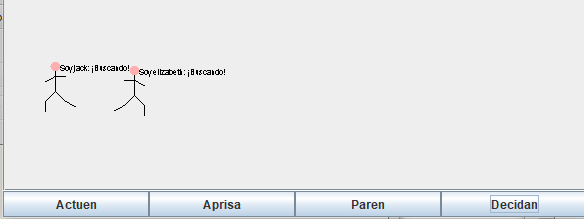
**¿Como quedan todos los piratas después de esta orden? Capture la pantalla inicial y la final.**

Quedan igual ya que no se ejecuta ninguna acción

****

1. **En este punto vamos a construir el método que atiende el click del botón Decidan de la interfaz: el método llamado decidan() de la clase Isla. Construya el método, ejecute el programa y haga click en el botón Decidan. ¿Como quedan todos los piratas después de esta orden? Capture la pantalla inicial y la final.**

Se mueven aleatoriamente



**Ciclo 2. Incluyendo a los Piratas rebeldes (NO OLVIDE BDD – MDD) (Construir: diseñar, implementar y probar)**

**El objetivo de este punto es permitir recibir en la isla Piratas rebeldes.**

**Los piratas rebeldes, vestidos de rojo, sólo actúan cada tres órdenes seguidas, cuando se les pide que paren, actuan y cuando se les pide que decidan, siempre paran si la última vez actuaron.**

**Adicionalmente, su mensaje es “Soy rebelde”.**

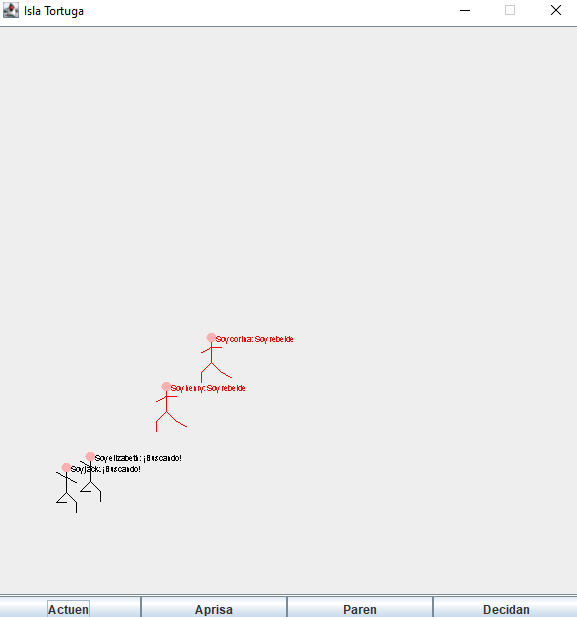
1. **Implemente este nuevo pirata. ¿cuáles métodos se sobre-escriben (overriding)?**

**Se implemento actue, pare y decida con las nuevas restricciones**

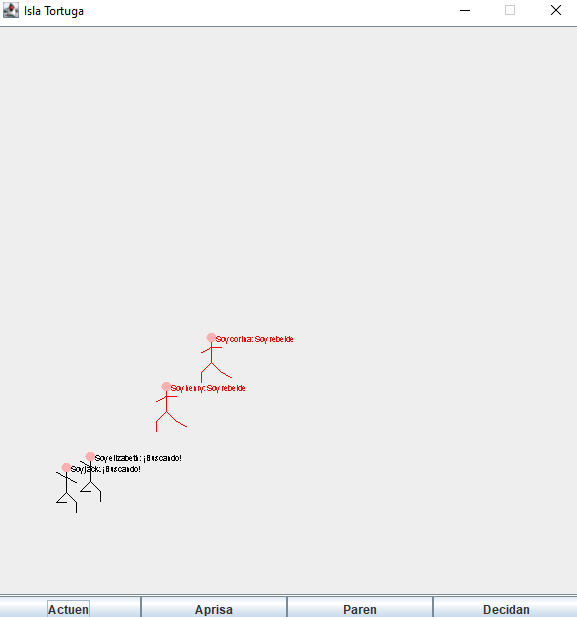
**2. Adicione una pareja de piratas necios, llámelos henry y corina, ejecute el programa y pídales a todos que actúen y que paren. Capture la pantalla.**

**¿Qué pasa?**

**Se comportan según las restricciones que se establecieron anteriormente**

****

1. **Nuevamente ejecute el programa y pídales a todos que actuen, decidan y paren. Capture la pantalla. ¿Qué pasa?**

****

**Cada uno de los piratas actúa según su comportamiento**

**Ciclo 3. Adicionando palmeras**

**El objetivo de este punto es incluir en la isla palmeras (sólo vamos a permitir un tipo de palmeras).**

**Las palmeras son verdes oscuras cuando actúan, verdes claras cuando paran y deciden siguiendo dos veces las reglas de la forma estándar.**

**Las palmeras son silenciosas. (NO OLVIDE BDD – MDD) (Construir: diseñar, implementar y probar)**

1. **¿Qué debemos hacer para tener palmeras en la isla?**

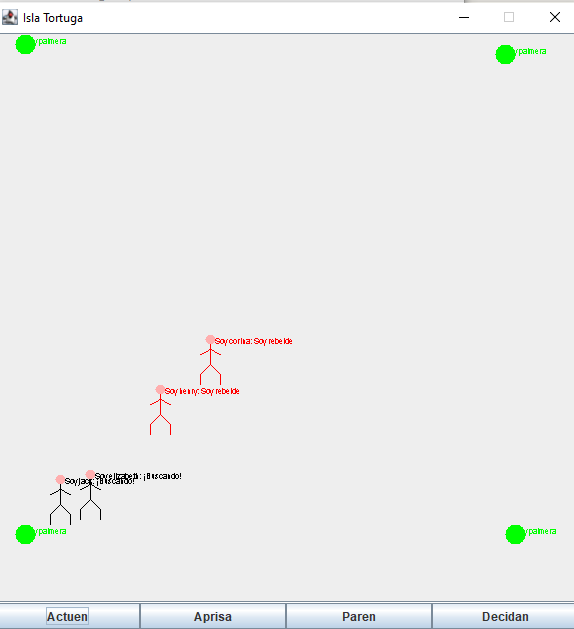
Implementar la clase con todos los métodos correspondientes

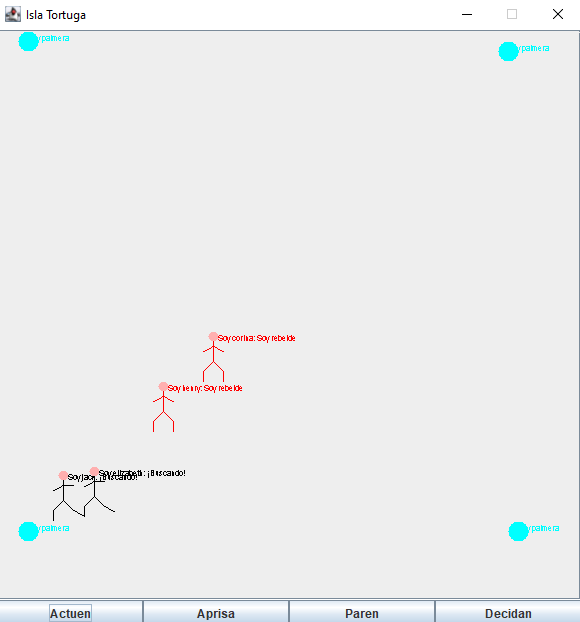
**2.Construya la clase Palmera para poder adicionarla en el Isla. ¿qué cambios incluyó Para aceptar este elemento , ¿debe cambiar en el código del Isla. en algo? ¿por qué?**

Se hace uso de las figuras que ya tiene la interfaz

**3. Adicionen cuatro palmeras en las esquinas del Isla, llámenlas superiorDerecha, superiorIzquierda, inferiorDerecha y inferiorIzquierda. Ejecuten el programa pídales a todos que actuen, decidan y paren.**

**Capturen la pantalla.**

****

****

**¿Qué pasa? ¿es correcto?**

Las palmeras actúan de manera correcta

**Ciclo 4. Creando un nuevo pirata: el minucioso objetivo de este punto es permitir recibir en la isla Piratas rebeldes.**

**El pirata minucioso debe hacer un recorrido tal que garantice que recorre todo el desierto, de modo que sin importar donde está el tesoro eventualmente pase por allí.**

**Está vestido con color naranja.**

**Cuando para, habla mucho.**

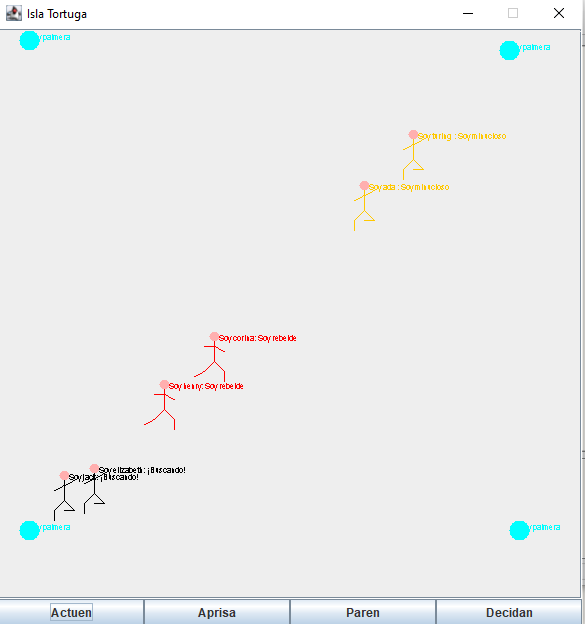
**(NO OLVIDE BDD – MDD) (Construir: diseñar, implementar y probar)**

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

**Se implemento actue, pare y decida con las nuevas restricciones**

**2. Adicione una pareja de piratas minuciosos, llámelos ada y turing, ejecute el programa y pídales a todos que actúen y que paren. Capture la pantalla.**

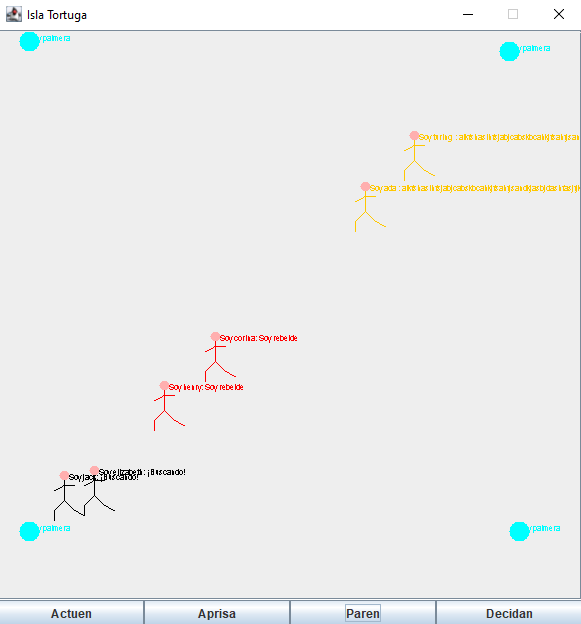
**¿Qué pasa?**

****

**3. Nuevamente ejecute el programa y pídales a todos que actúen, decidan y paren. Capture la pantalla.**

**¿Qué pasa?**

**Actúan según las restricciones que se establecieron**

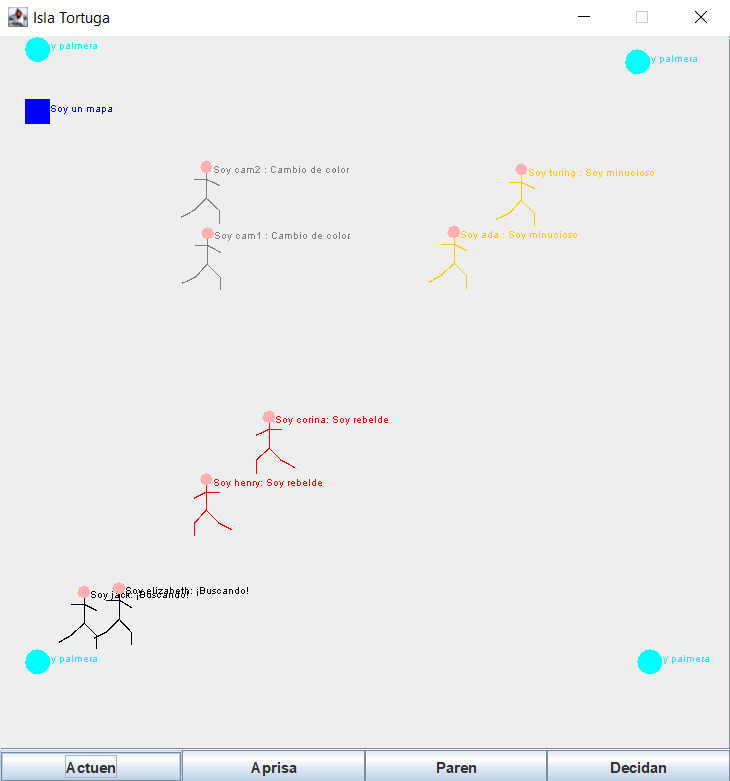
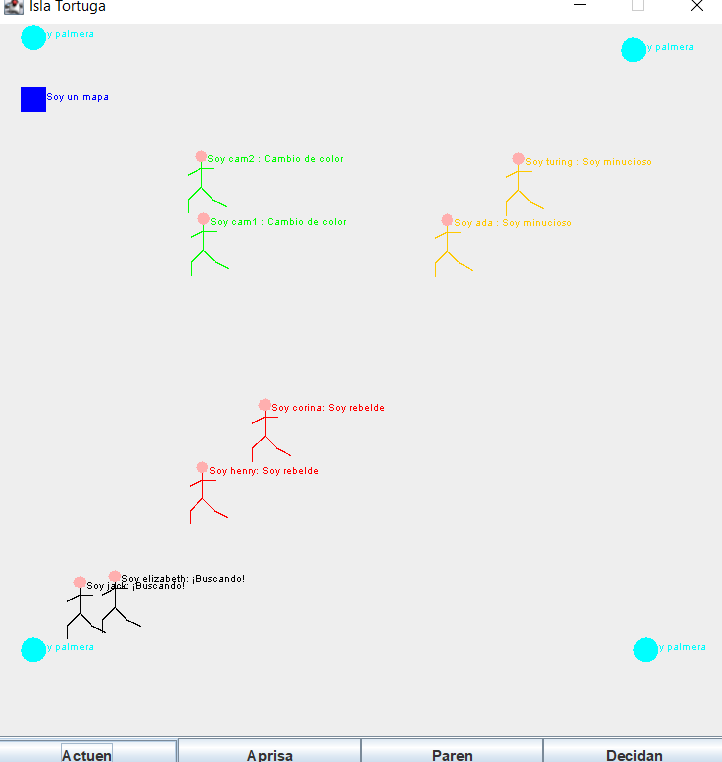
****

**Ciclo 5. Nuevo pirata: Proponiendo y diseñando El objetivo de este punto es permitir recibir en un nuevo tipo de pirata. (NO OLVIDE BDD - MDD)**

**1.Propongan, describan e implementen un nuevo tipo de pirata.**

**Se implementa un pirata camaleón, que al actuar cambia de color aleatoriamente**

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

** **

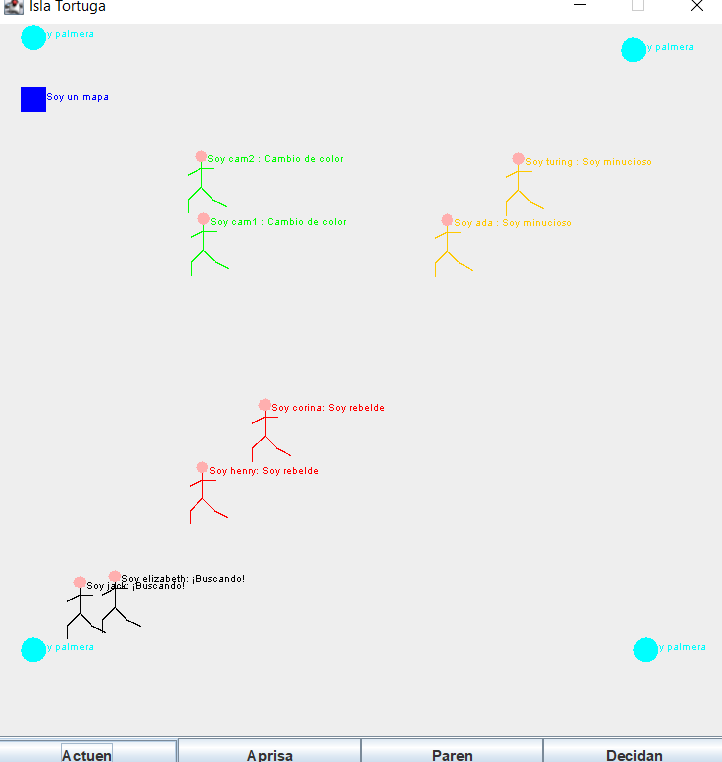
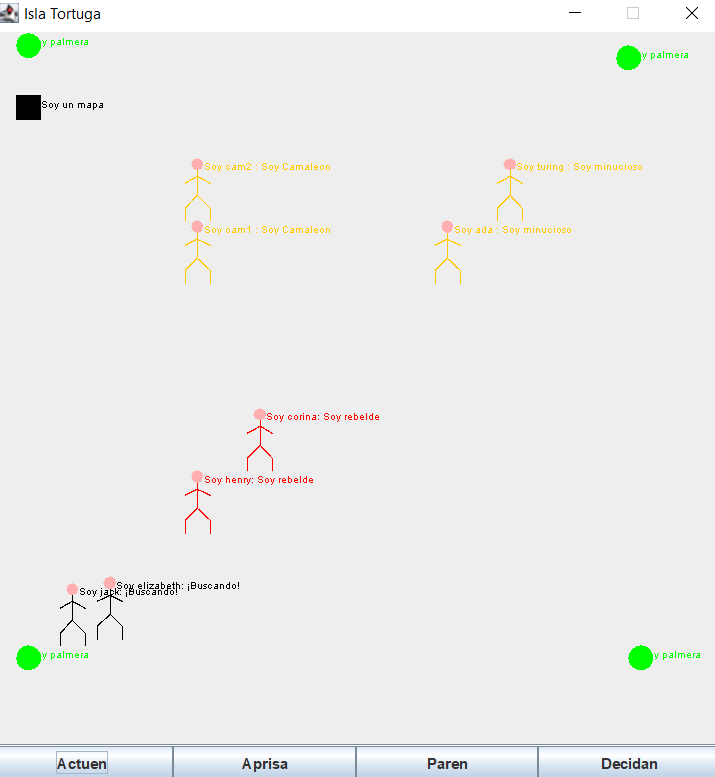
**Ciclo 6. Nuevo elemento: Proponiendo y diseñando El objetivo de este punto es permitir recibir en un nuevo elemento en la isla (NO OLVIDE BDD - MDD)**

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

**La isla va a tener mapas que ayuden a ubicar a los piratas, cuando los piratas actúan, el mapa cambia de color para facilitar que lo vean**

**2. Incluyan dos de ellos con los nombres semánticos.**

**Ejecuten el programa con dos casos significativos. Explique la intención de cada caso y capture las pantallas correspondientes.**

** **

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

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

**2. Consulte el comando java para ejecutar un archivo jar. Ejecutelo**

**¿qué pasa?**

**Se ejecuta la aplicación**

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

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

**2. 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 IslaTortuga src aplicacion presentacion pruebas**

**3. En el directorio copien únicamente los archivos \*.java del paquete de aplicación . Consulte y capture el contenido de src/aplicacion Estructura de proyectos java [En lab03.doc] En java los proyectos se estructuran considerando tres directorios básicos. automata src bin docs**

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

**2. ¿Qué archivos debería copiar del proyecto original al directorio bin? ¿Por qué? Cópielos y consulte y capture el contenido del directorio que modificó. Comandos de java [En lab03.doc]**

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

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

**3. Busque la opción que sirve para conocer la versión a que corresponden estos dos comandos. Documente el resultado. Compilando [En lab03.doc]**

**1. Utilizando el comando javac, desde el directorio raiz (desde automata con una sóla 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.**

**2. Revise de nuevo el contenido del directorio de trabajo y sus subdirectorios. ¿Cuáles nuevos archivos aparecen ahora y dónde se ubican? Documentando [En lab03.doc]**

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

**2. ¿Cuál archivo hay qué abrir para empezar a navegar por la documentación? Ábralo y capture la pantalla. Ejecutando [En lab03.doc]**

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

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

**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. Ejecute 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**](http://junit.sourceforge.net/doc/cookbook/cookbook.htm)

**3. Pegue en su documento el resultado de las pruebas Empaquetando**

**1. Consulte 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 ?**

**2. ¿Cómo se ejecuta el proyecto empaquetado?**

**RETROSPECTIVA**

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

**14 horas**

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

**En todos los métodos hizo falta su diagrama de secuencias y falto documentar el código.**

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

**El mayor logro lograr implementar todos los métodos.**

**El mayor problema fue implementar el diagrama de paquetes ya que no entendía cómo se debía diseñar.**

**Para resolverlo investigamos en internet.**

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

**Me comprometo a dedicar más tiempo en los laboratorios para alcanzar a elaborar todas las partes del laboratorio.**