

6PROGRAMACIÓN ORIENTADA A OBJETOS

Persistencia

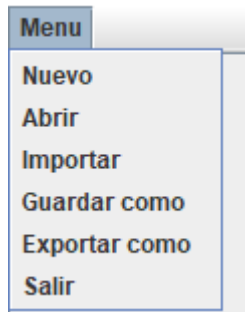
2021-02 Laboratorio 6/6 [:)]

Creando la maqueta

1. MODELO: Preparen en la clase fachada del modelo los métodos correspondientes a las cuatro opciones básicas de entrada-salida (abra, guarde, importe y exporte). Los métodos deben simplemente propagar una ArticoExcepcion con el mensaje genérico de “Opción en construcción”. Los métodos deben tener un parámetro File.

```
public Artico abra(File archivo)throws ArticoExcepcion {  
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);  
}  
  
public Artico guarde(File archivo)throws ArticoExcepcion{  
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);  
}  
  
public Artico importe(File archivo)throws ArticoExcepcion{  
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);  
}  
  
public Artico exporte(File archivo)throws ArticoExcepcion{  
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);  
}
```

2. VISTA : Construyan un menú barra que ofrezca, además de las opciones básicas de entrada-salida, las opciones estándar de nuevo y salir (Nuevo, Abrir, Guardar como, Importar, Exportar como, Salir). No olviden incluir los separadores. Para esto creen el método prepareElementosMenu. Únicamente debe funcionar la vista. Capturen la pantalla correspondiente.



3. CONTROLADOR: Construyan los oyentes correspondientes a las seis opciones. Para esto creen el método `prepareAccionesMenu` y los métodos base del controlador (`opcionAbrir`, `opcionGuardar`, `opcionExportar`, `opcionImportar`, `opcionNuevo`, `opcionSalir`). Estos métodos, por ahora, llaman directamente el método correspondiente de la capa de dominio. No incluyan todavía el `FileChooser`. Capturen una pantalla significativa.

```
private void prepareAccionesMenu(){
    ActionListener oyenteBotonNuevo=new ActionListener(){
        public void actionPerformed(ActionEvent e){
        }
    };
    nuevo.addActionListener(oyenteBotonNuevo);

    ActionListener oyenteBotonAbrir=new ActionListener(){
        public void actionPerformed(ActionEvent e){
            try {
                Artico.demeArtico().abra(new File( pathname: ""));
            } catch (ArticoExcepcion ex) {
                System.out.println(ex.getMessage());
            }
        }
    };
    abrir.addActionListener(oyenteBotonAbrir);

    ActionListener oyenteBotonImportar=new ActionListener(){
        public void actionPerformed(ActionEvent e){
            try {
                Artico.demeArtico().importe(new File( pathname: ""));
            } catch (ArticoExcepcion ex) {
                System.out.println(ex.getMessage());
            }
        }
    };
    importar.addActionListener(oyenteBotonImportar);

    ActionListener oyenteBotonGuardarComo=new ActionListener(){
        public void actionPerformed(ActionEvent e){
            try {
                Artico.demeArtico().guarde(new File( pathname: ""));
            } catch (ArticoExcepcion ex) {
                System.out.println(ex.getMessage());
            }
        }
    };
    guardarComo.addActionListener(oyenteBotonGuardarComo);
}
```

```

        ActionListener oyenteBotonExportarComo=new ActionListener(){
            public void actionPerformed(ActionEvent e){
                try {
                    Artico.demeArtico().exporte(new File( pathname: ""));
                } catch (ArticoExcepcion ex) {
                    System.out.println(ex.getMessage());
                }
            }
        };
        exportarComo.addActionListener(oyenteBotonExportarComo);

        ActionListener oyenteBotonSalir=new ActionListener(){
            public void actionPerformed(ActionEvent e){
                |
            }
        };
        salir.addActionListener(oyenteBotonSalir);
    }

```

Implementando salir y nuevo

1. Construyan el método opcionSalir que hace que se termine la aplicación. No es necesario incluir confirmación.

```

        ActionListener oyenteBotonSalir=new ActionListener(){
            public void actionPerformed(ActionEvent e){
                System.exit( status: 0);
            }
        };
        salir.addActionListener(oyenteBotonSalir);
    }

```

2. Construyan el método `opcionNuevo` que crea un nuevo artículo. Capturen una pantalla significativa.

```
private void prepareAccionesMenu(){
    ActionListener oyenteBotonNuevo=new ActionListener(){
        public void actionPerformed(ActionEvent e){
            Artico.demeArtico().nuevo();
        }
    };
    nuevo.addActionListener(oyenteBotonNuevo);
}
```

```
public void nuevo(){
    nuevoArtico();
}
```

Implementando salvar y abrir

1. Copien las versiones actuales de `abra` y `guarde` y renómbrenlos como `abra00` y `guarde00`

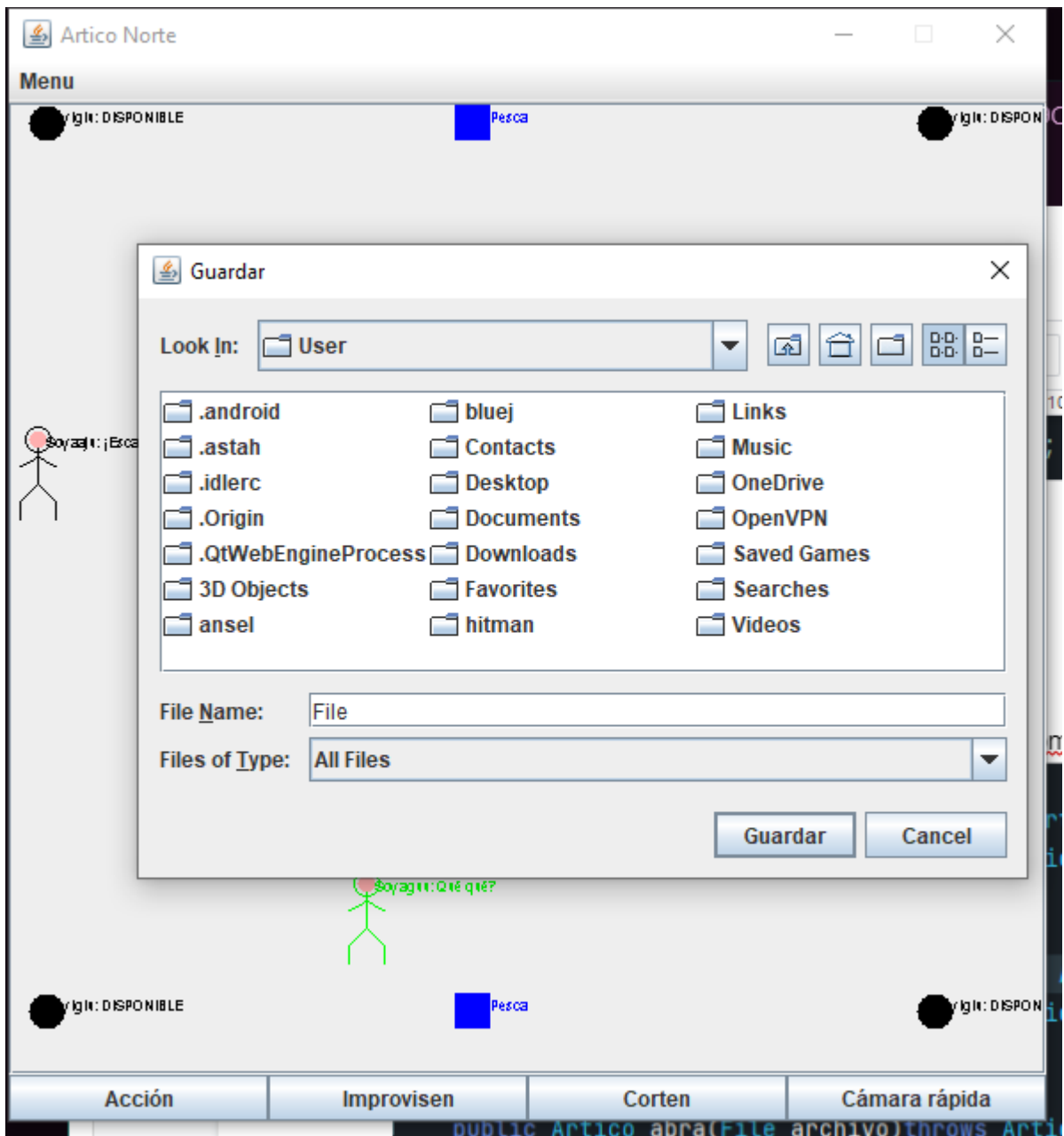
```
public Artico abra00(File archivo)throws ArticoExcepcion {
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);
}

public Artico guarde00(File archivo)throws ArticoExcepcion{
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);
}

public Artico abra(File archivo)throws ArticoExcepcion {
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);
}

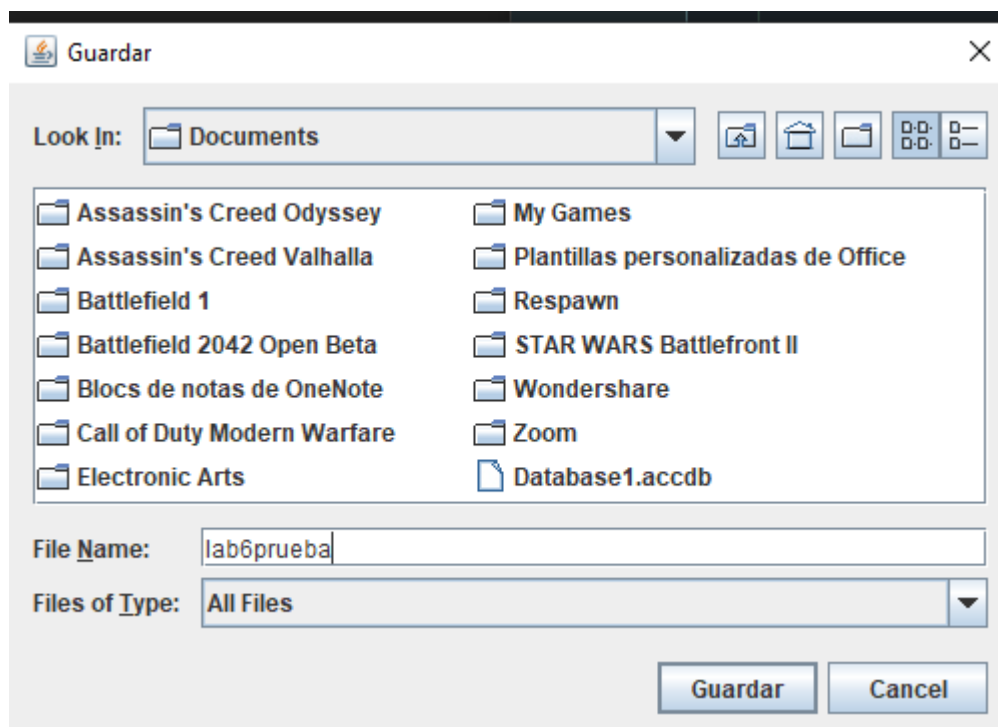
public Artico guarde(File archivo)throws ArticoExcepcion{
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);
}
```

2. Construyan el método `opcionGuardar` que une de forma adecuada la capa de presentación con la capa de dominio. Usen un `FileChooser` y atiendan la excepción. Ejecuten la aplicación probando las diferentes opciones del `FileChooser` y capturen una pantalla significativa.

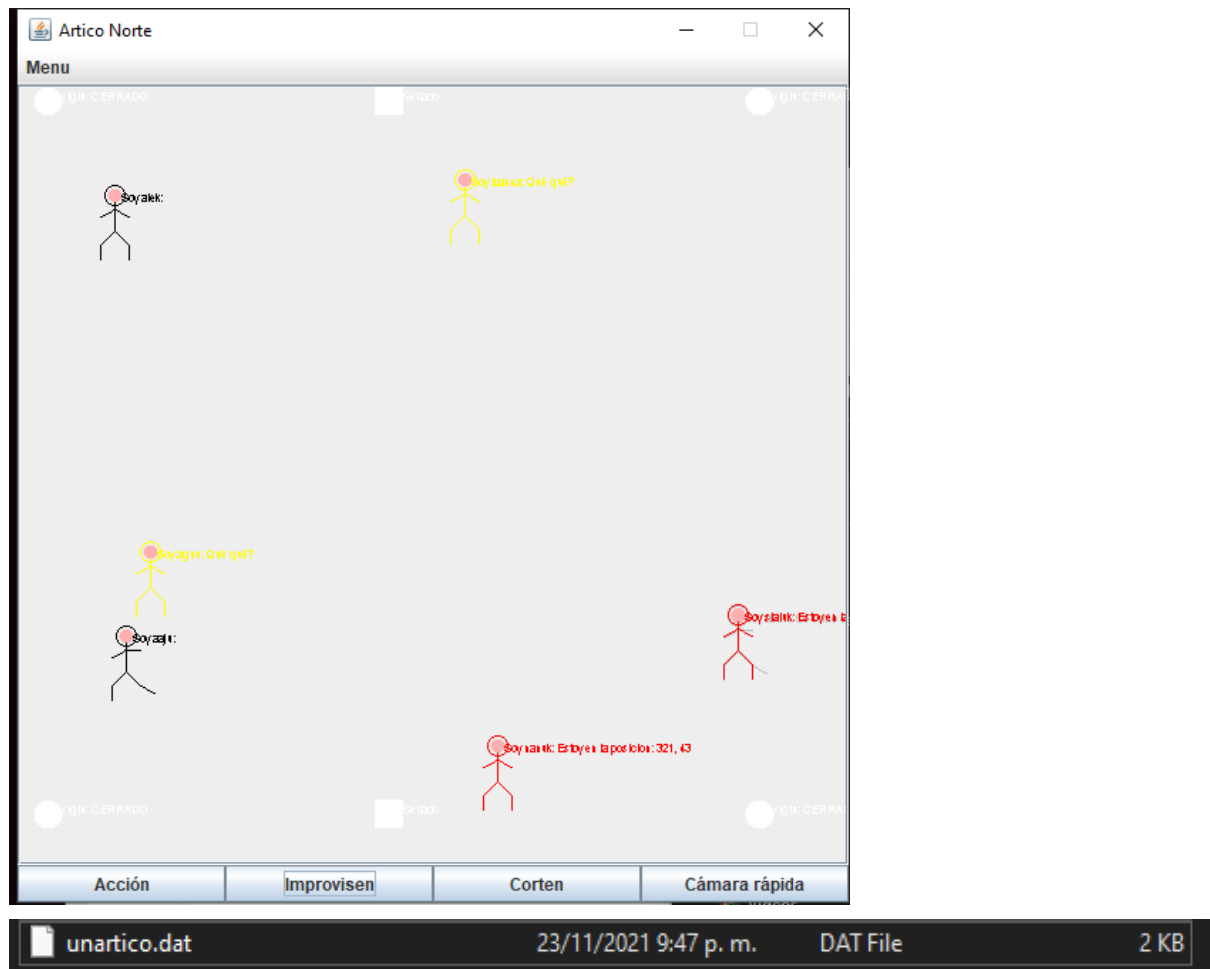


3. Construyan el método `guarde` que ofrece el servicio de guardar en un archivo el estado actual del artículo.

```
public void guarde00(File archivo) throws ArticoExcepcion {
    ObjectOutputStream salida = null;
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(archivo);
        salida = new ObjectOutputStream(fos);
        salida.writeObject(this);
    } catch (IOException ex) {
        System.out.println("Fallo la cerrar del archivo");
    }
}
```

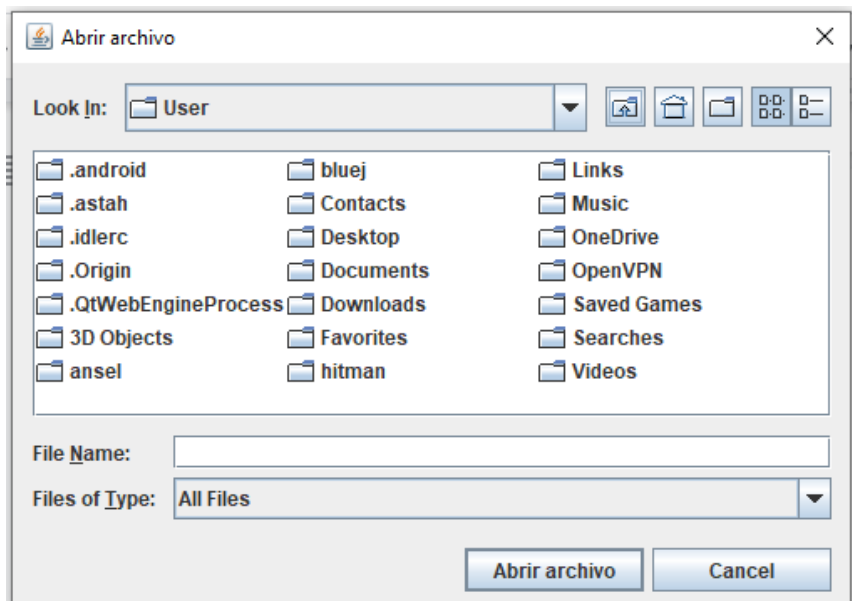


4. Validen este método guardando la isla inicial después de dos clics como `unaartico.dat`.
¿El archivo se creó en el disco? ¿Cuánto espacio ocupa?



Location: D:\UN\POOB\tercio3\lab6
Size: 1,49 KB (1.529 bytes)
Size on disk: 4,00 KB (4.096 bytes)

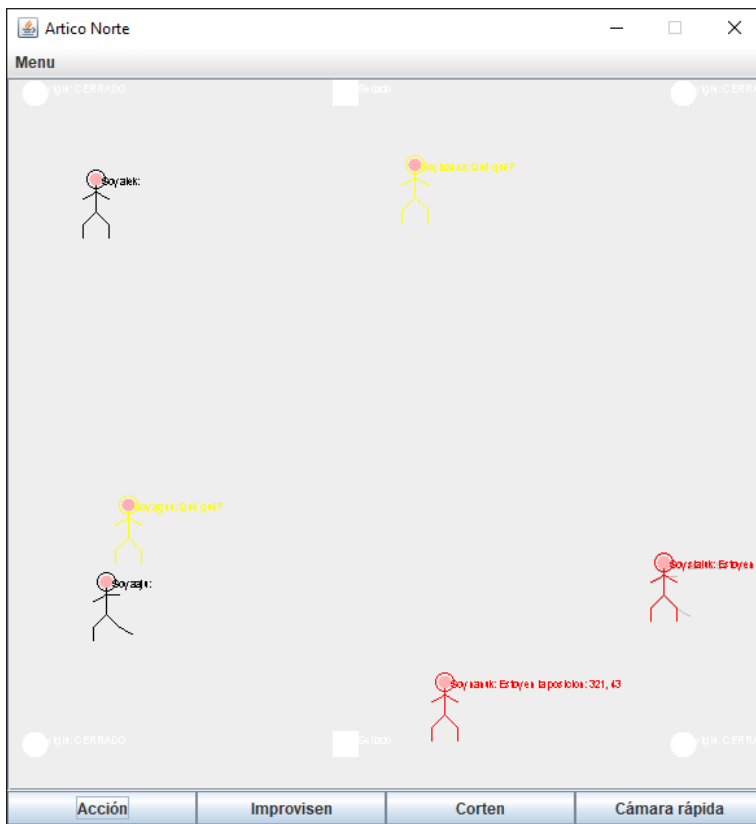
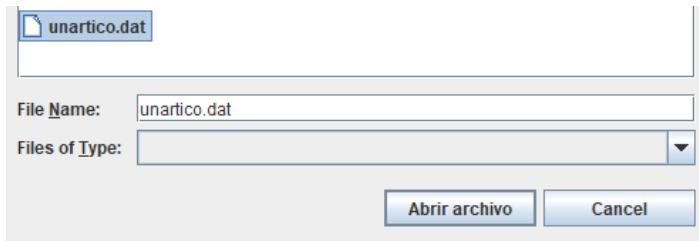
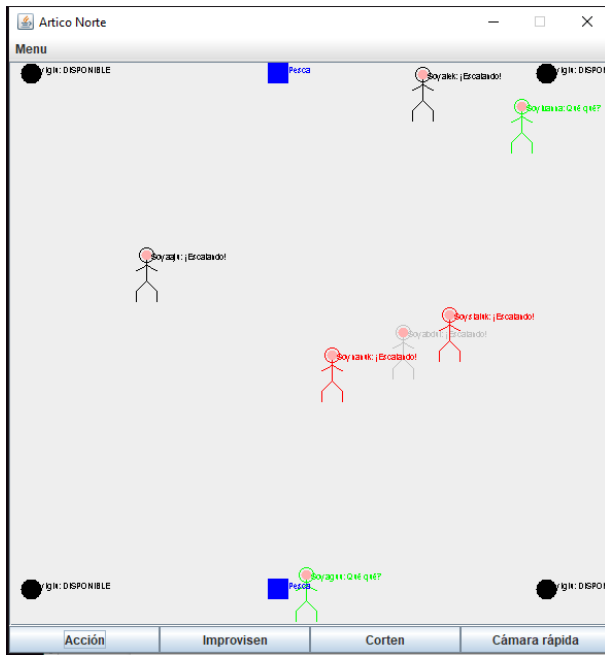
5. Construyan el método `opcionAbrir` que une de forma adecuada la capa de presentación con la capa de dominio. Ejecuten la aplicación probando las diferentes opciones del `FileChooser` y capturen una pantalla significativa.



6. Construyan el método `abra` que ofrece el servicio de leer un artículo de un archivo. Por ahora para las excepciones sólo consideren un mensaje de error general.

```
public Artico abra00(File archivo) throws ArticoExcepcion {
    try {
        ObjectInputStream archivoArtico = new ObjectInputStream(new FileInputStream(archivo));
        Artico articoCargado = (Artico) archivoArtico.readObject();
        archivoArtico.close();
        cambieArtico(articoCargado);
        return articoCargado;
    } catch (IOException | ClassNotFoundException e) {
        throw new ArticoExcepcion(ArticoExcepcion.NO_ABRI0);
    }
}
```

7. Realicen una prueba de aceptación para este método iniciando la aplicación, creando una nueva situación en el artículo y abriendo el archivo `unaartico.dat`. Capturen imágenes significativas de estos resultados.

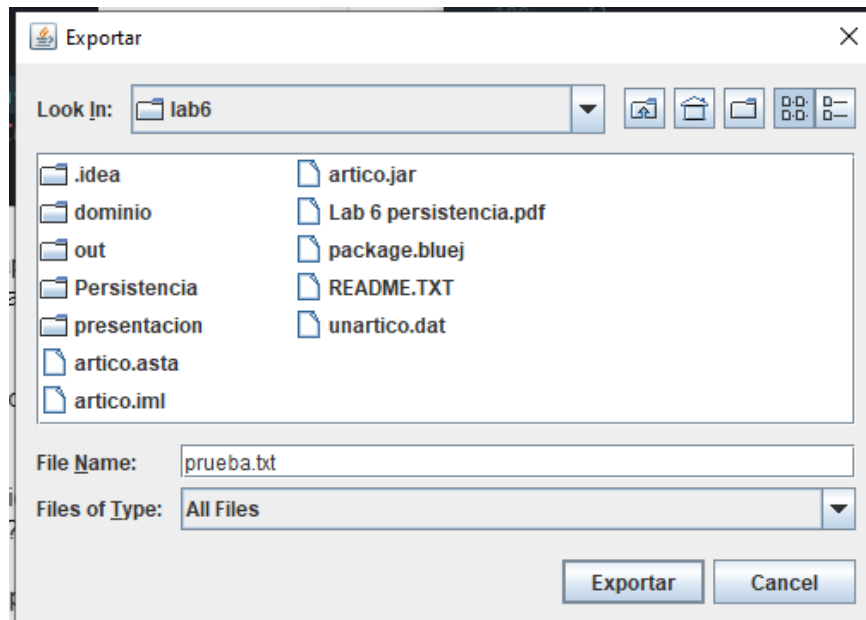
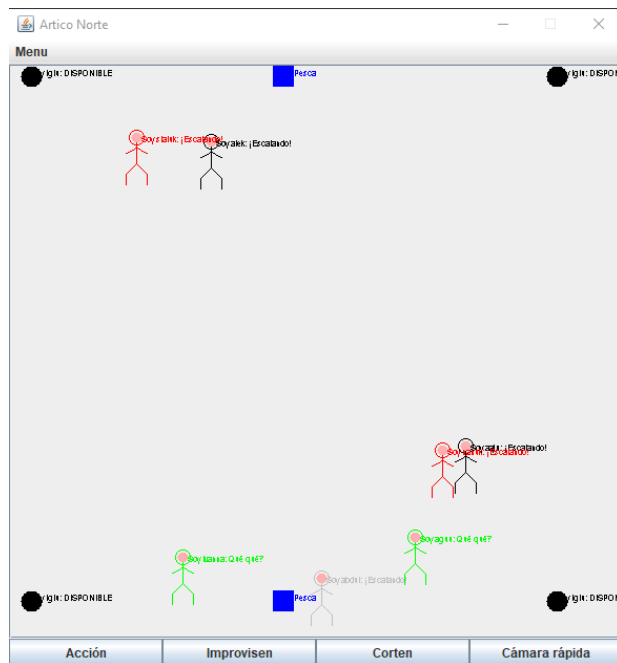


Implementando importar y exportar

1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe00 y exporte00

```
public Artico importe00(File archivo)throws ArticoExcepcion{  
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);  
}  
  
public Artico exporte00(File archivo)throws ArticoExcepcion{  
    throw new ArticoExcepcion(ArticoExcepcion.EN_CONSTRUCCION);  
}
```

2. Construyan el método opcionExportar que une de forma adecuada la capa de presentación con la capa de dominio. Ejecuten la aplicación y capturen una pantalla significativa.



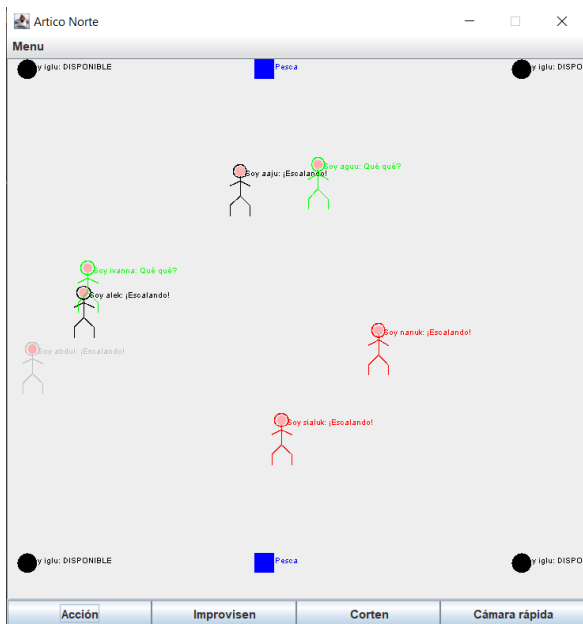
3. Construyan el método exporte que ofrece el servicio de exportar a un archivo texto, con el formato definido, el estado actual.

```

public void exporte00(File archivo) throws ArticoExcepcion{
    try{
        PrintWriter printWriter = new PrintWriter(new FileOutputStream(archivo));
        for(EnArtico elemento : elementos){
            printWriter.println(elemento.getClass().getSimpleName() + " " + elemento.getName() + " " + elemento.getPosicionX() + " " + elemento.getPosicionY() + "\n");
        }
        printWriter.close();
    } catch (FileNotFoundException e) {
        throw new ArticoExcepcion(ArticoExcepcion.NO_EXPORTO);
    }
}

```

4. Realicen una prueba de aceptación de este método: iniciando la aplicación y exportando como unaartico.txt. Editen el archivo y analicen los resultados. ¿Qué pasó?



```

unaartico - Notepad
File Edit Format View Help
Esquimal aaju 220 392

Esquimal alek 62 269

EsquimalSordo aguu 299 399

EsquimalSordo ivanna 66 294

EsquimalExperto abdul 10 212

EsquimalExplorador nanuk 360 231

EsquimalExplorador sialuk 262 140

HuecoPesca pescaArriba 250 500

HuecoPesca pescaAbajo 250 0

Iglu superiorDerecha 500 500

Iglu superiorIzquierda 0 500

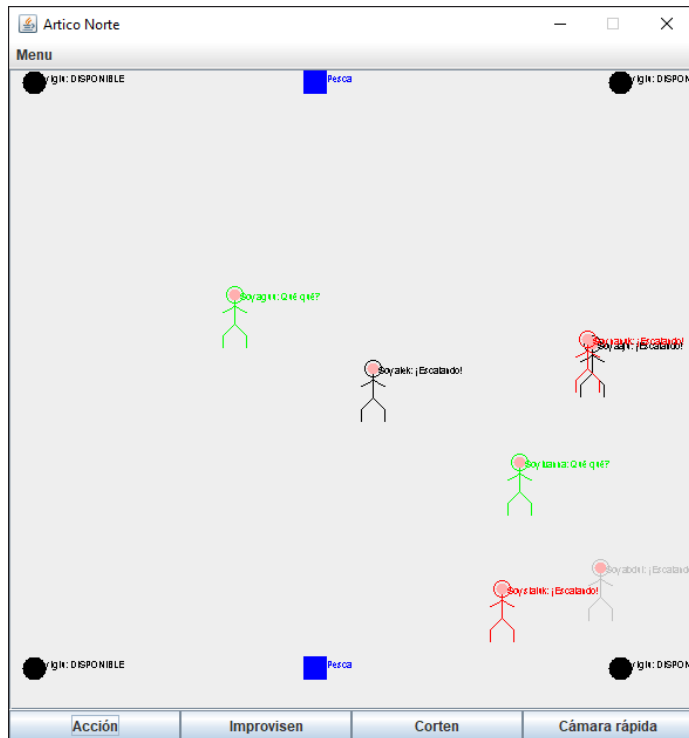
Iglu inferiorIzquierda 0 0

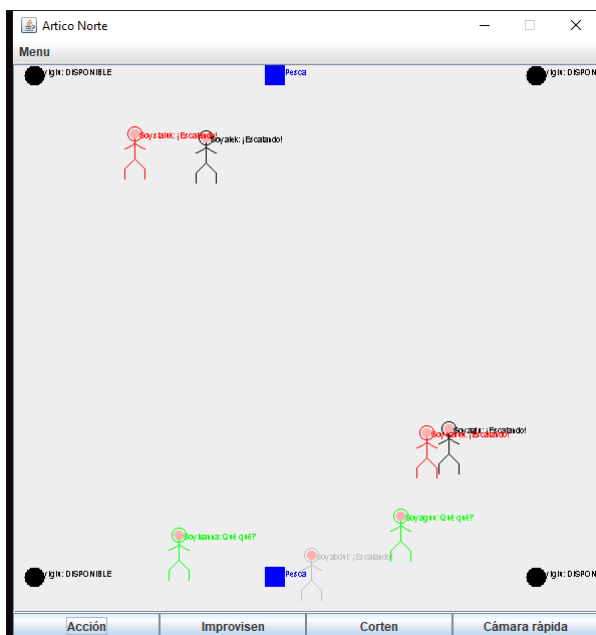
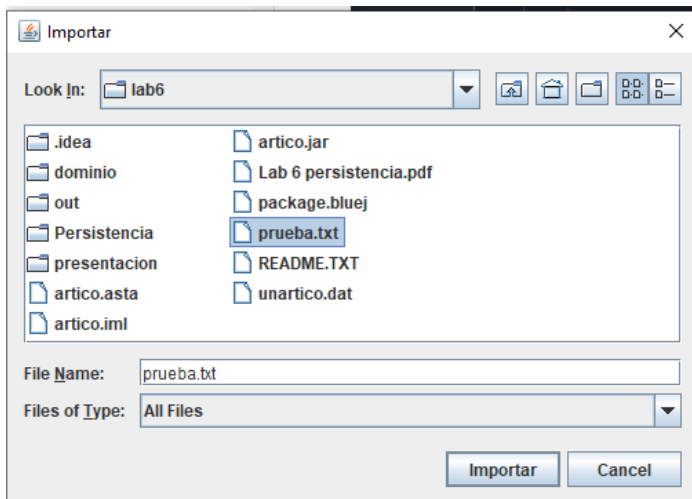
Iglu inferiorDerecha 500 0

```

como aun no tenemos implementado el metodo de importar, no se va a ver ningun cambio

5. Construyan el método `opcionImportar` que une de forma adecuada la capa de presentación con la capa de dominio. Ejecuten la aplicación y capturen una pantalla significativa.

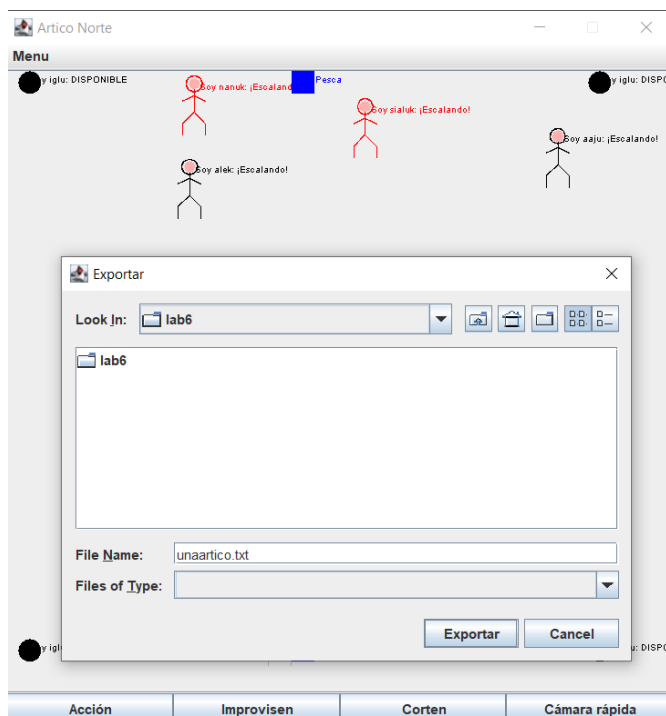


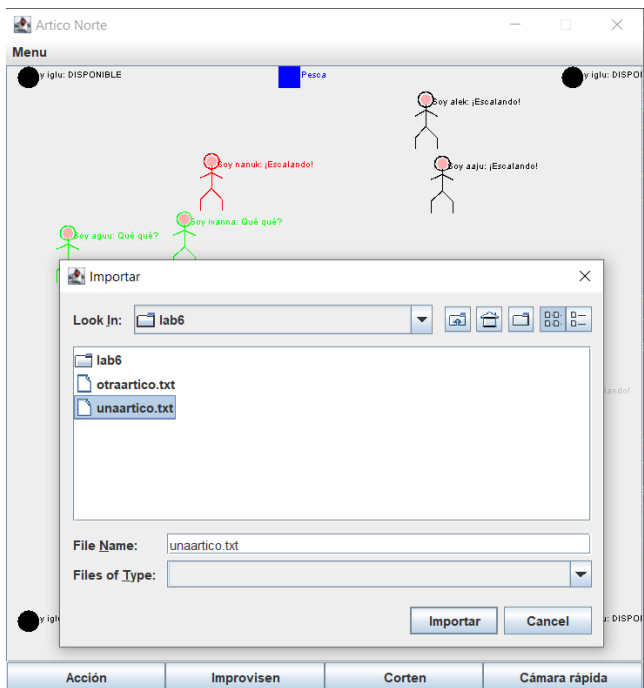
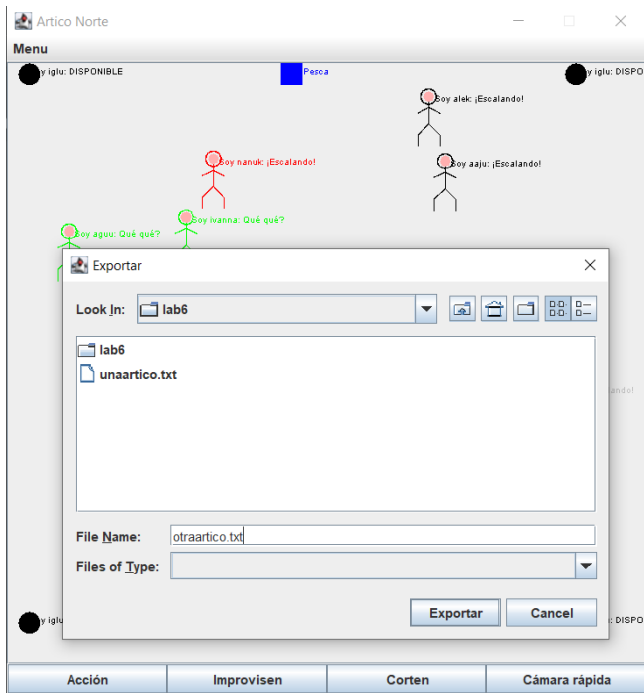


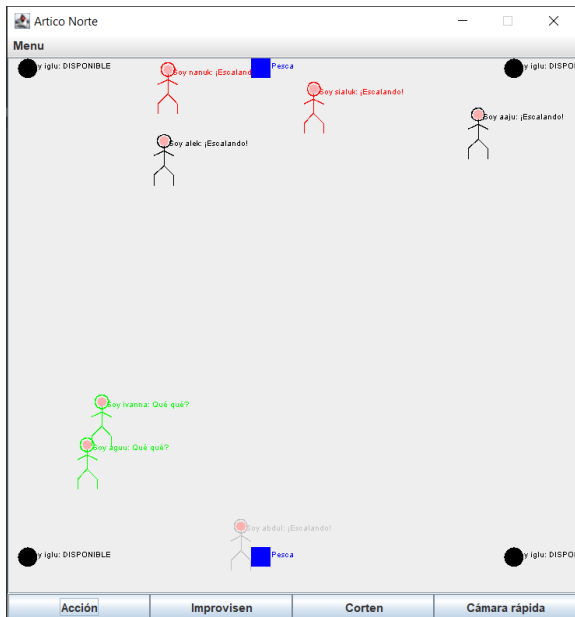
6. Construyan el método importe que ofrece el servicio de importar de un archivo texto con el formato definido. Por ahora sólo considere un mensaje de error general. (Consulten en la clase String los métodos trim y split)

```
public Articulo importe(File archivo) throws ArticuloException {
    try {
        nuevoArticulo();
        BufferedReader bin = new BufferedReader(new FileReader(archivo));
        String line = bin.readLine();
        while (line != null) {
            line = line.trim();
            if (line.equals("")) continue;
            String[] elements = line.split("\\s+");
            if (elements[0].equals("Esquimal")) {
                Esquimal esquimal = new Esquimal(demeArticulo(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimal);
                adiciona(esquimal);
            }
            else if (elements[0].equals("EsquimalSordo")) {
                EsquimalSordo esquimalSordo = new EsquimalSordo(demeArticulo(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimalSordo);
                adiciona(esquimalSordo);
            }
            else if (elements[0].equals("EsquimalExperto")) {
                EsquimalExperto esquimalExperto = new EsquimalExperto(demeArticulo(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimalExperto);
                adiciona(esquimalExperto);
            }
            else if (elements[0].equals("EsquimalExplorador")) {
                EsquimalExplorador esquimalExplorador = new EsquimalExplorador(demeArticulo(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]), esquimales);
                esquimales.add(esquimalExplorador);
                adiciona(esquimalExplorador);
            }
            else if (elements[0].equals("Iglu")) {
                Iglu iglu = new Iglu(demeArticulo(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                adiciona(iglu);
            }
            else if (elements[0].equals("HuecoPesca")) {
                HuecoPesca huecoPesca = new HuecoPesca(demeArticulo(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                adiciona(huecoPesca);
            }
            line = bin.readLine();
        }
        return this;
    } catch (IOException g) {
        throw new ArticuloException(ArticuloException.NO_EXPORTO);
    }
}
```

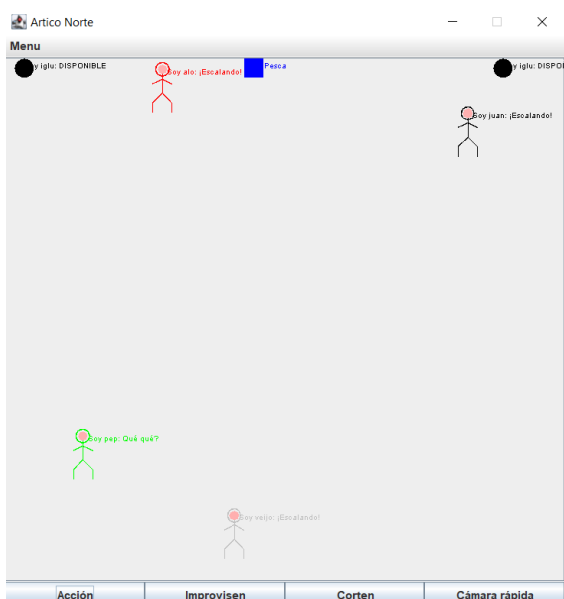
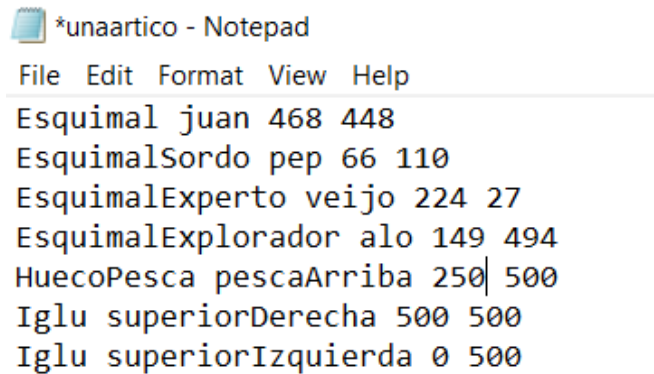
7. Realicen una prueba de aceptación de este par de métodos: iniciando la aplicación exportando a unaartico.txt. saliendo, entrando, creando un nuevo artico e importando el archivo otraartico.txt. ¿Qué resultado obtuvieron? Capturen la pantalla final.





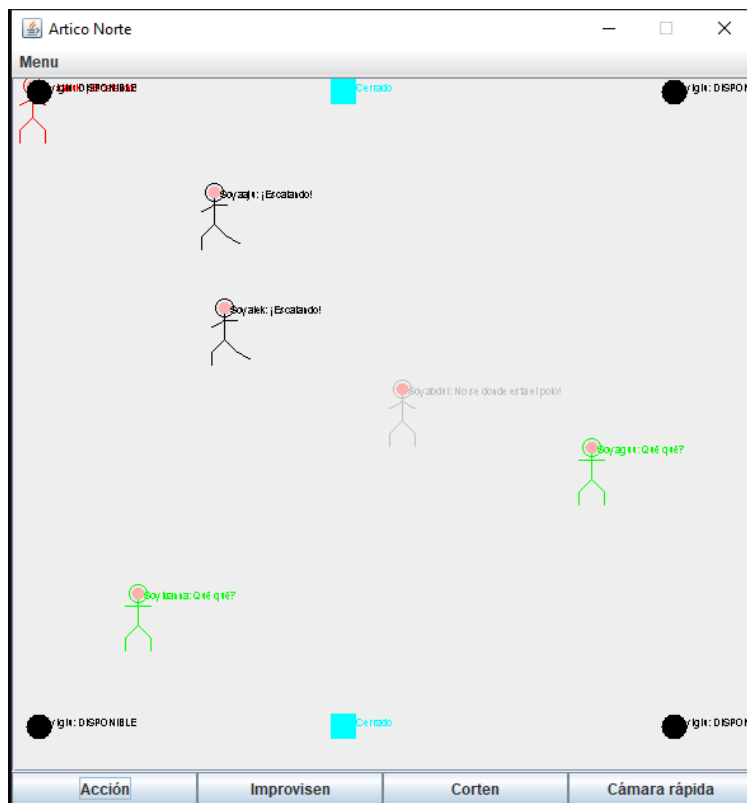
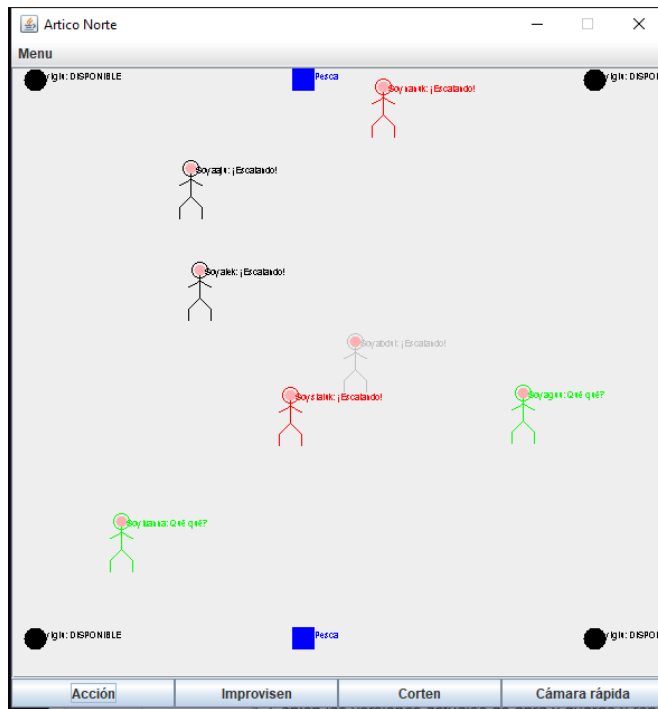


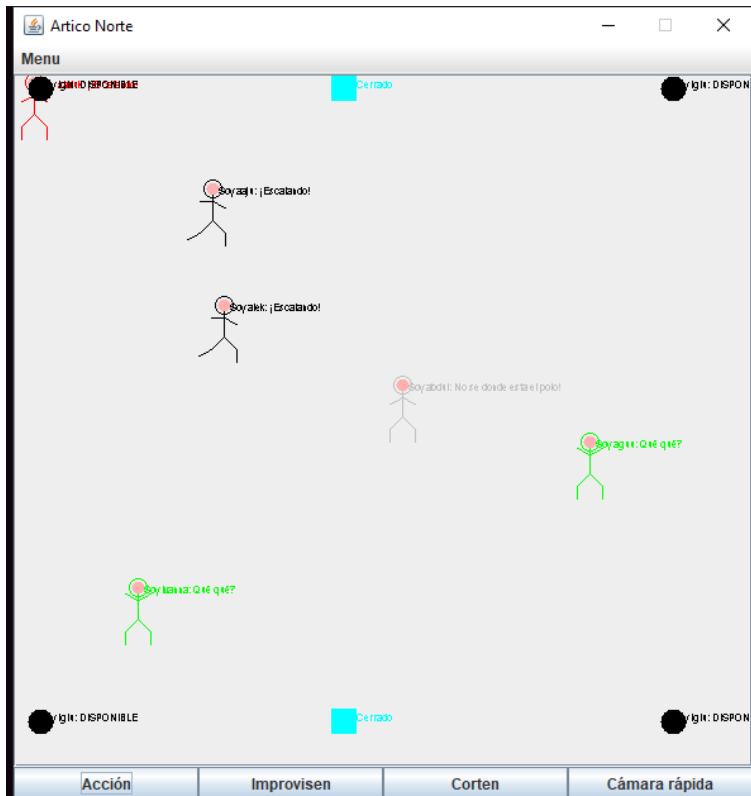
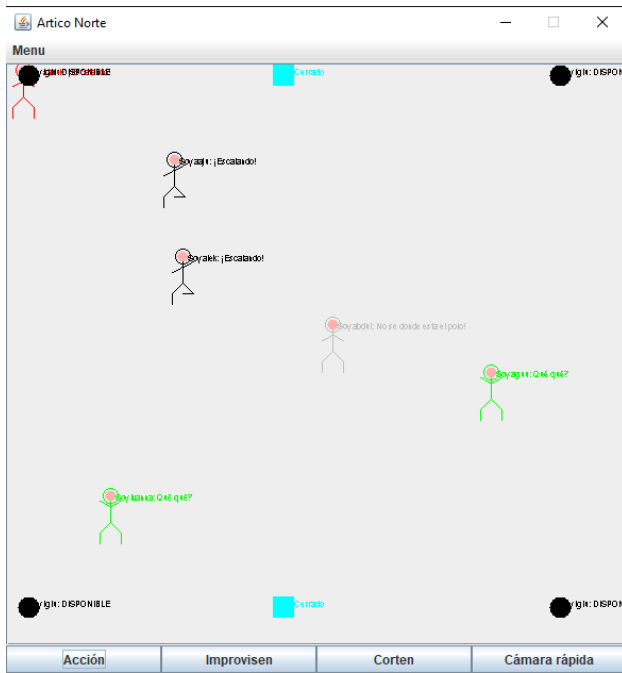
8. Realicen otra prueba de aceptación de este método escribiendo un archivo de texto correcto en unaartico.txt. e importe este archivo. ¿Qué resultado obtuvieron? Capturen la pantalla.

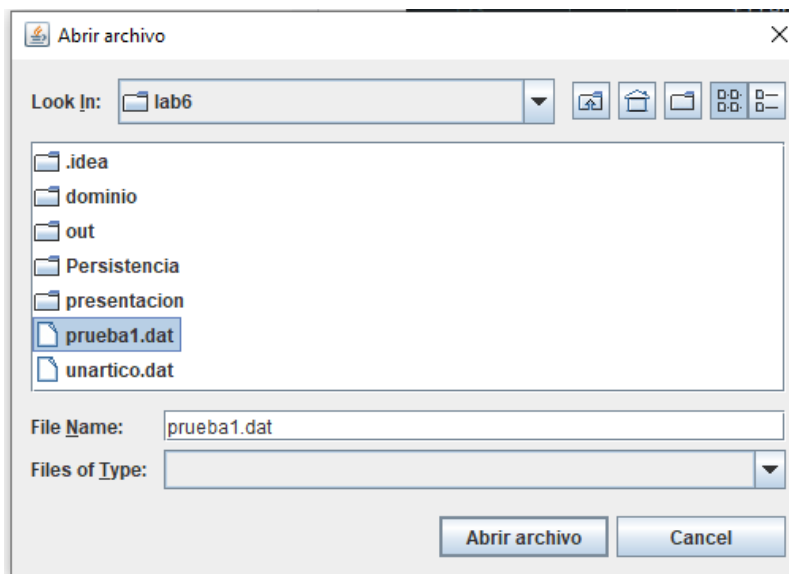
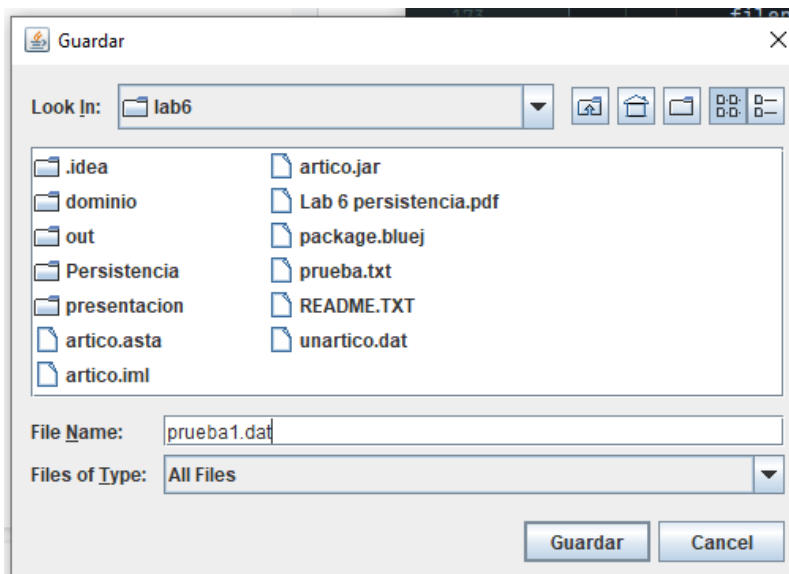


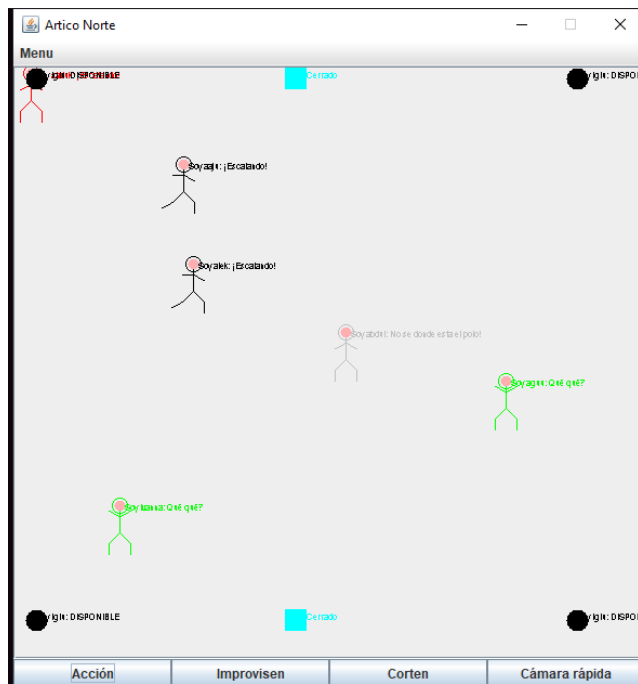
Analizando comportamiento

1. Ejecuten la aplicación, den tres clics, guarden a un archivo cualquiera y ábralo.
Describan el comportamiento



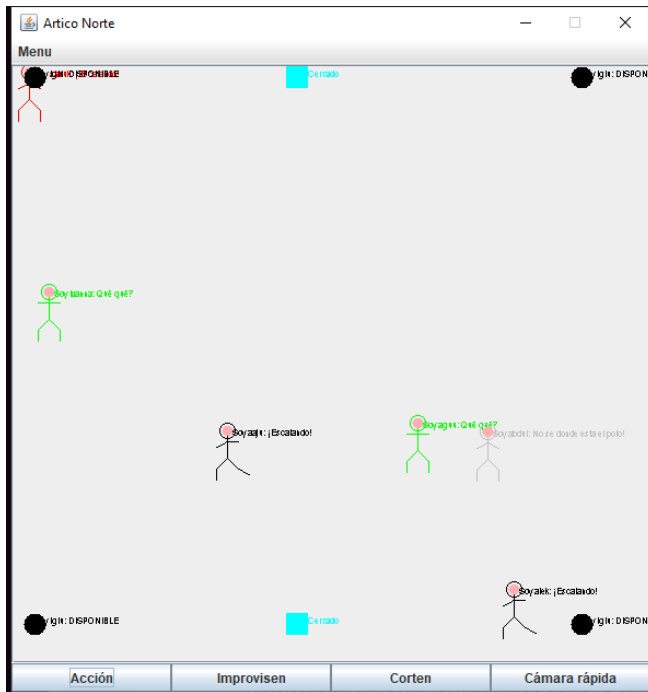
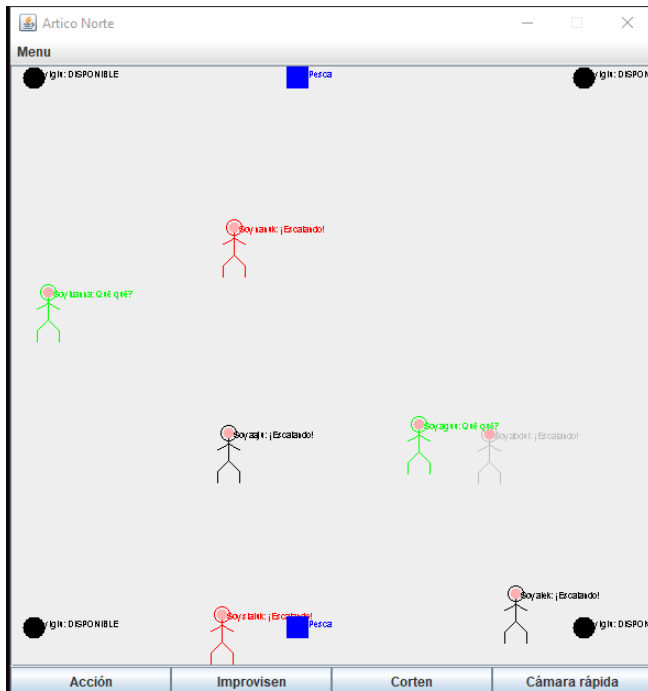


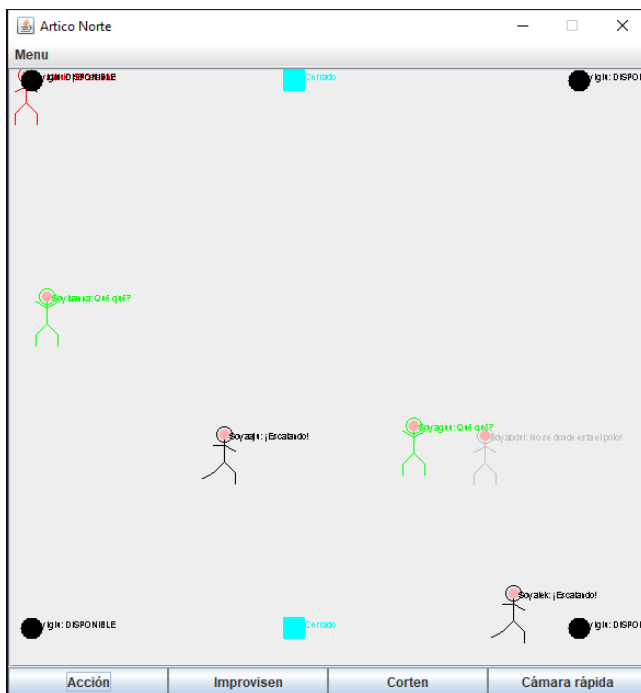
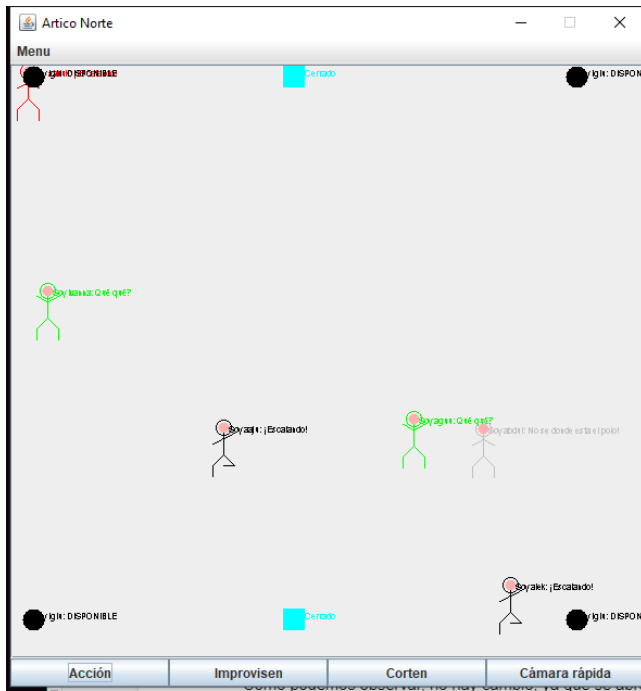


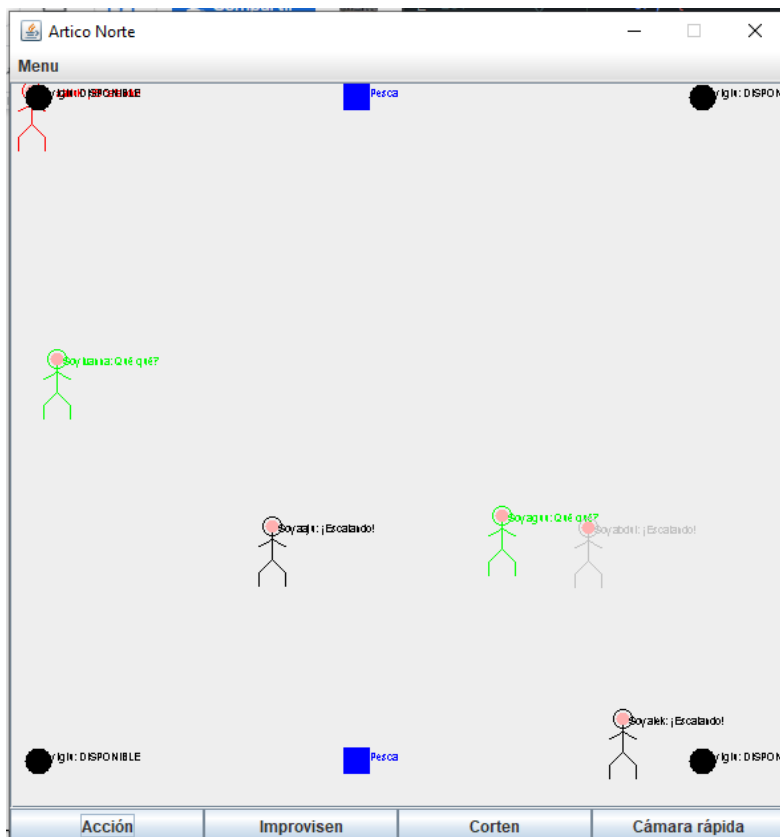
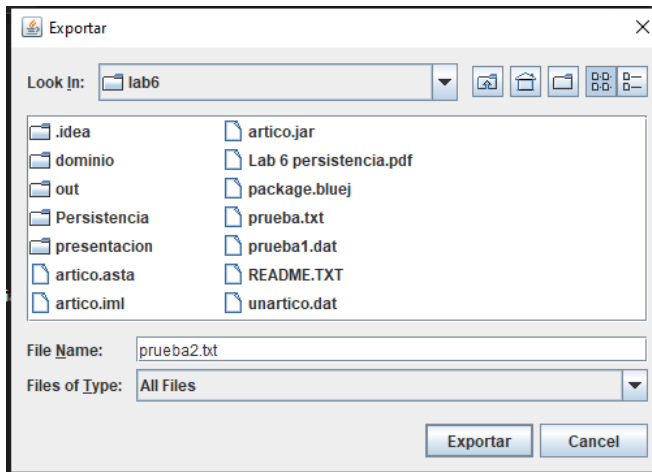


Como podemos observar, no hay cambio, ya que se abre el mismo archivo que se guarda.

2. Ejecuten la aplicación, tres clics, exporten a un archivo cualquiera e importen. Describan el comportamiento







3. ¿Qué diferencias ven el comportamiento 1. y 2.? Expliquen los resultados.

La diferencia es que en 1 si se mantienen las acciones que se realizaron, mientras que en 2 no.

Perfeccionando salvar y abrir

1. Copien las versiones actuales de abra y guarde y renómbrenlos como abra01 y guarde01


```

public Artico abra01(File archivo) throws ArticoExcepcion {
    try {
        ObjectInputStream archivoArtico = new ObjectInputStream(new FileInputStream(archivo));
        Artico articoCargado = (Artico) archivoArtico.readObject();
        archivoArtico.close();
        cambieArtico(articoCargado);
        return articoCargado;
    } catch (IOException | ClassNotFoundException e) {
        throw new ArticoExcepcion(ArticoExcepcion.NO_ABRIO);
    }
}

public void guarde01(File archivo) throws ArticoExcepcion {
    ObjectOutputStream salida = null;
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(archivo);
        salida = new ObjectOutputStream(fos);
        salida.writeObject(this);
    } catch (IOException ex) {
        throw new ArticoExcepcion(ArticoExcepcion.NO_GUARDO);
    }
}

```

2. Perfeccionen el manejo de excepciones de los métodos abra y guarde detallando los errores.

```

public Artico abra(File archivo) throws ArticoExcepcion {
    try {
        ObjectInputStream archivoArtico = new ObjectInputStream(new FileInputStream(archivo));
        Artico articoCargado = (Artico) archivoArtico.readObject();
        archivoArtico.close();
        cambieArtico(articoCargado);
        return articoCargado;
    } catch (FileNotFoundException e) {
        throw new ArticoExcepcion( ArticoExcepcion.ARCHIVO_NO_ENCONTRADO);
    } catch (StreamCorruptedException e) {
        throw new ArticoExcepcion( ArticoExcepcion.ARCHIVO_CORRUPTO);
    } catch (ClassNotFoundException e) {
        throw new ArticoExcepcion( ArticoExcepcion.CLASE_NO_ENCONTRADA);
    } catch (InvalidClassException e) {
        throw new ArticoExcepcion( ArticoExcepcion.CLASE_NO_PERMITIDA);
    } catch (OptionalDataException e) {
        throw new ArticoExcepcion( ArticoExcepcion.DATOS_PRIMITIVOS);
    } catch (IOException e) {
        throw new ArticoExcepcion( ArticoExcepcion.ENTRADA_SALIDA);
    } catch (Exception e) {
        e.printStackTrace();
        throw new ArticoExcepcion( ArticoExcepcion.NO_ABRIO);
    }
}

```

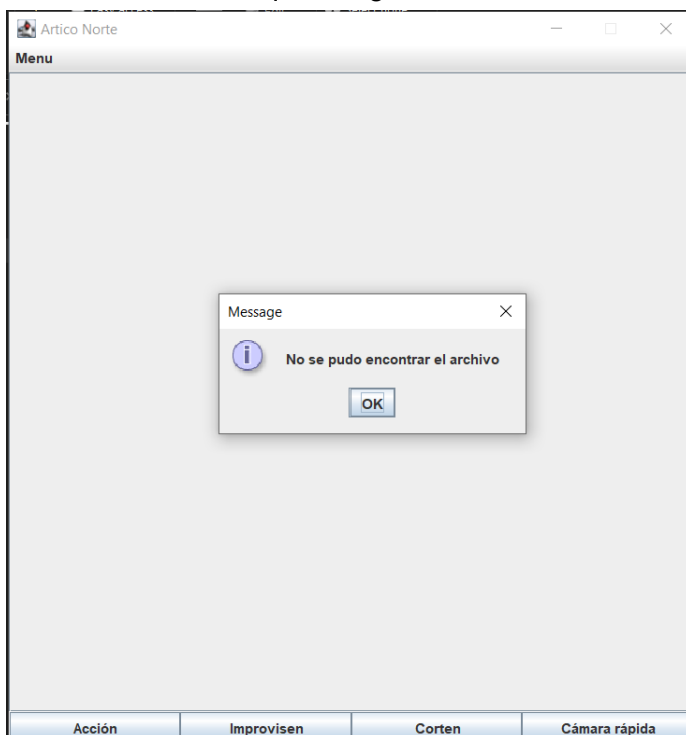
```

public void guarde(File archivo) throws ArticoExcepcion{
    ObjectOutputStream salida = null;
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(archivo);
        salida = new ObjectOutputStream(fos);
        salida.writeObject(this);
    } catch (IOException ex) {
        throw new ArticoExcepcion(ArticoExcepcion.NO_GUARDO);
    }
}

```

3. Realicen una prueba de aceptación para validar uno de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final.

editamos el archivo que fue guardado



Perfeccionando importar y exportar.

1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe01 y exporte01

```

public Artico _importe01(File _archivo)throws ArticoExcepcion{
    try{
        nuevoArtico();
        BufferedReader bIn = new BufferedReader(new FileReader(_archivo));
        String _line = bIn.readLine();
        while (_line != null) {
            _line = _line.trim();
            if (_line.equals("\n")) continue;
            String[] elements = _line.split(" ");
            if (elements[0].equals("Esquimal")) {
                Esquimal esquimal = new Esquimal(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimal);
                adicione(esquimal);
            }
            else if (elements[0].equals("EsquimalSordo")) {
                EsquimalSordo esquimalSordo = new EsquimalSordo(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimalSordo);
                adicione(esquimalSordo);
            }
            else if (elements[0].equals("EsquimalExperto")) {
                EsquimalExperto esquimalExperto = new EsquimalExperto(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimalExperto);
                adicione(esquimalExperto);
            }
            else if (elements[0].equals("EsquimalExplorador")) {
                EsquimalExplorador esquimalExplorador = new EsquimalExplorador(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]), esquimal);
                esquimales.add(esquimalExplorador);
                adicione(esquimalExplorador);
            }
            else if (elements[0].equals("Iglu")) {
                Iglu iglu = new Iglu(demeArtico(), _name, Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                adicione(iglu);
            }
            else if (elements[0].equals("HuecoPesca")) {
                HuecoPesca huecoPesca = new HuecoPesca(demeArtico(), _name, Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                adicione(huecoPesca);
            }
            _line = bIn.readLine();
        }
        return this;
    } catch (IOException e) {
        throw new ArticoExcepcion(ArticoExcepcion.NO_EXPORTO);
    }
}

```

```

public void _exporte01(File _archivo)throws ArticoExcepcion{
    try{
        PrintWriter printWriter = new PrintWriter(new FileOutputStream(_archivo));
        for(EnArtico elemento : elementos){
            printWriter.println(elemento.getClass().getSimpleName() + " " + elemento.getName() + " " + elemento.getPosicionX() + " " + elemento.getPosicionY() + "\n");
        }
        printWriter.close();
    } catch (FileNotFoundException e) {
        throw new ArticoExcepcion(ArticoExcepcion.NO_EXPORTO);
    }
}

```

2. Perfeccionen el manejo de excepciones de los métodos importe y exporte detallando los errores.

```

        }
        return this;
    } catch (ArticoExcepcion e) {
        throw new ArticoExcepcion(ArticoExcepcion.CLASE_NO_ENCONTRADA);
    } catch (Exception e) {
        e.printStackTrace();
        throw new ArticoExcepcion(ArticoExcepcion.NO_IMPORTO);
    }
}

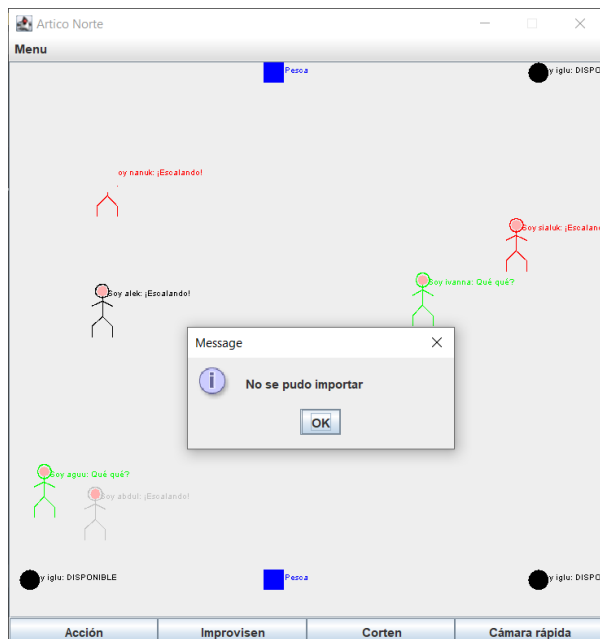
```

```

public void _exporte(File _archivo)throws ArticoExcepcion{
    try{
        PrintWriter printWriter = new PrintWriter(new FileOutputStream(_archivo));
        for(EnArtico elemento : elementos){
            printWriter.println(elemento.getClass().getSimpleName() + " " + elemento.getName() + " " + elemento.getPosicionX() + " " + elemento.getPosicionY());
        }
        printWriter.close();
    } catch (FileNotFoundException e) {
        throw new ArticoExcepcion(ArticoExcepcion.ARCHIVO_NO_ENCONTRADO);
    } catch (Exception e){
        throw new ArticoExcepcion(ArticoExcepcion.NO_EXPORTO);
    }
}

```

3. Realicen una prueba de aceptación para validar uno de los nuevos mensajes diseñados, ejecútenla y capturen la pantalla final.



Perfeccionando importar. Hacia un minicompilador

1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe02 y exporte02

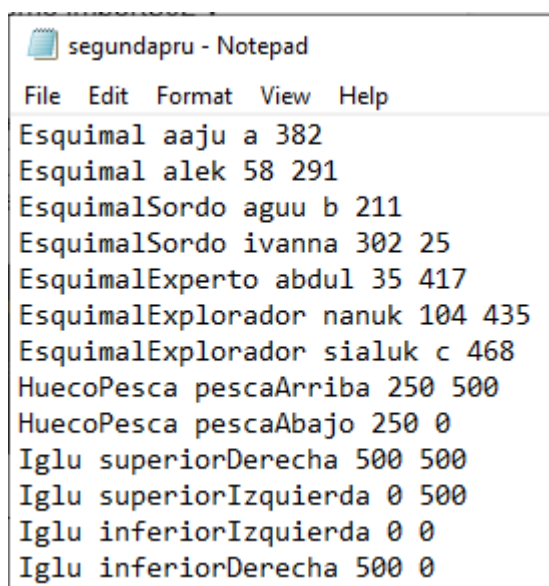
```
public Artico importe02(File archivo) throws ArticoException {
    try {
        nuevoArtico();
        BufferedReader bIn = new BufferedReader(new FileReader(archivo));
        String line = bIn.readLine();
        while (line != null) {
            line = line.trim();
            if (line.equals("\n")) continue;
            String[] elements = line.split(" ");
            if (elements[0].equals("Esquimal")) {
                Esquimal esquimal = new Esquimal(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimal);
                adicione(esquimal);
            } else if (elements[0].equals("EsquimalSordo")) {
                EsquimalSordo esquimalSordo = new EsquimalSordo(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimalSordo);
                adicione(esquimalSordo);
            } else if (elements[0].equals("EsquimalExperto")) {
                EsquimalExperto esquimalExperto = new EsquimalExperto(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                esquimales.add(esquimalExperto);
                adicione(esquimalExperto);
            } else if (elements[0].equals("EsquimalExplorador")) {
                EsquimalExplorador esquimalExplorador = new EsquimalExplorador(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]), esquimales);
                esquimales.add(esquimalExplorador);
                adicione(esquimalExplorador);
            } else if (elements[0].equals("Iglu")) {
                Iglu iglu = new Iglu(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                adicione(iglu);
            } else if (elements[0].equals("HuecoPesca")) {
                HuecoPesca huecoPesca = new HuecoPesca(demeArtico(), elements[1], Integer.parseInt(elements[2]), Integer.parseInt(elements[3]));
                adicione(huecoPesca);
            } else {
            }
            line = bIn.readLine();
        }
    } catch (IOException e) {
        throw new ArticoException("Error al importar el archivo: " + e.getMessage());
    }
}

public void exporte02(File archivo) throws ArticoException {
    try {
        PrintWriter printWriter = new PrintWriter(new FileOutputStream(archivo));
        for (EnArtico elemento : elementos) {
            printWriter.println(elemento.getClass().getSimpleName() + " " + elemento.getName() + " " + elemento.getPosicionX() + " " + elemento.getPosicionY() + "\n");
        }
        printWriter.close();
    } catch (FileNotFoundException e) {
        throw new ArticoException(ArticoException.ARCHIVO_NO_ENCONTRADO);
    } catch (Exception e) {
        throw new ArticoException(ArticoException.NO_EXPORTO);
    }
}
```

2. Perfeccionen el método importe para que, además de los errores generales, en las excepciones indique el detalle de los errores encontrados en el archivo (como un compilador) : número de línea donde se encontró el error, palabra que tiene el error y causa de error.

```
private static void escribirError(String error, int linea, int posicion, File file) throws IOException{
    FileWriter fw = new FileWriter(file, append: true);
    fw.write( str: "se encontro que " + error + " en la linea " + linea + " en la palabra " + posicion + "\n");
    fw.close();
}
```

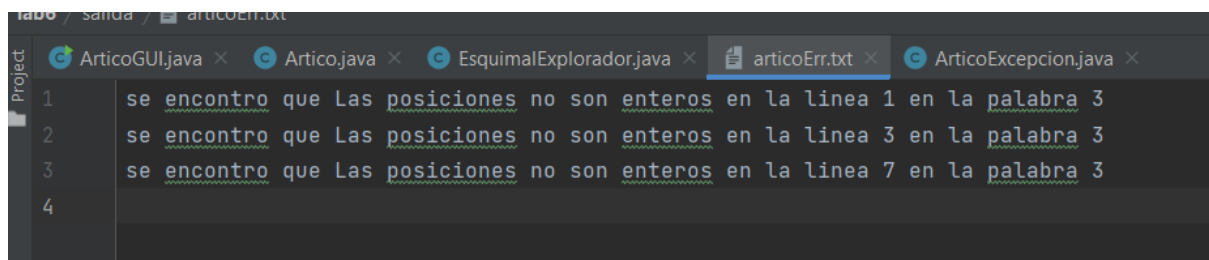
3. Escriban otro archivo con errores, llámelo articoErr.txt, para ir arreglándolo con ayuda de su “importador”. Presente las pantallas que contengan los errores.



segundapru - Notepad

File Edit Format View Help

Esquimal aaju a 382
Esquimal alek 58 291
EsquimalSordo aguu b 211
EsquimalSordo ivanna 302 25
EsquimalExperto abdul 35 417
EsquimalExplorador nanuk 104 435
EsquimalExplorador sialuk c 468
HuecoPesca pescaArriba 250 500
HuecoPesca pescaAbajo 250 0
Iglu superiorDerecha 500 500
Iglu superiorIzquierda 0 500
Iglu inferiorIzquierda 0 0
Iglu inferiorDerecha 500 0



labo / salida / articoErr.txt

Project

ArticoGUI.java × Artico.java × EsquimalExplorador.java × articoErr.txt × ArticoExcepcion.java ×

1 se encontro que Las posiciones no son enteros en la linea 1 en la palabra 3
2 se encontro que Las posiciones no son enteros en la linea 3 en la palabra 3
3 se encontro que Las posiciones no son enteros en la linea 7 en la palabra 3
4

Perfeccionando importar. Hacia un minicompilador flexible.

1. Copien las versiones actuales de importe y exporte y renómbrenlos como importe03 y exporte03

```

public Artico importe03(File archivo) throws ArticoExcepcion {
    try {
        ArrayList<String> lines = new ArrayList<>();
        int cont = 0;
        nuevoArtico();
        BufferedReader bIn = new BufferedReader(new FileReader(archivo));
        String line = bIn.readLine();
        while (line != null) {
            lines.add(line);
            line = bIn.readLine();
        }
        verificaErrores(lines);
        Artico a = new Artico();
        for(String ln: lines){
            String[] lineclean;
            lineclean = ln.trim().split(regex: " ");
            int fila = Integer.parseInt(lineclean[2]);
            int columna = Integer.parseInt(lineclean[3]);
            if(!Objects.equals(lineclean[0], b: "EsquimalExplorador")) {
                System.out.println("entro a primer class for name");
                Class cl = Class.forName("dominio." + lineclean[0]);
                Class[] parameterType = {Artico.class, String.class, int.class, int.class};
                polo.adicione((EnArtico) cl.getDeclaredConstructor(parameterType).newInstance( _initargs: a, "Hola", fila, columna));
            } else{
                System.out.println("entro a segundo class for name");
                Class cl = Class.forName("dominio." + lineclean[0]);
                Class[] parameterType = {Artico.class, String.class, int.class, int.class, ArrayList.class};
                polo.adicione((EnArtico) cl.getDeclaredConstructor(parameterType).newInstance( _initargs: a, "Hola", fila, columna, esquimales));
            }
        }
        return this;
    } catch (ArticoExcepcion e) {

```

2. Perfeccionen los métodos importe y exporte para que pueda servir para cualquier tipo de elementos creados en el futuro (Investiguen cómo crear un objeto de una clase dado su nombre)

3. Escriban otro archivo de pruebas, llámelo articoErrG.txt, para probar la flexibilidad. Presente las pantallas que contenga un error significativo.

RETROSPECTIVA

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

Archila: 13 horas

Otero: 13 horas

2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?

satisfactorio, se completó casi en su totalidad el laboratorio, por que el último punto no es muy claro y no supimos qué hacer

3. Considerando la práctica XP del laboratorio ¿por qué consideran que es importante?

porque así el diseño y codificación quedan mucho más limpios y claros, tanto para nosotros, como para quien revise el programa

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

el mayor logro fue usar correctamente el classforname

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

complementar las dudas del otro, realizar pruebas mas fuertes