

MANUAL DEL PROGRAMADOR

NORBHEY DANILO MUÑOZ CAÑON
BRAYAN LEONARDO SIERRA FORERO
CARLOS DAVID CAMACHO PANCE



UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS
PROYECTO CURRICULAR INGENIERA DE SISTEMAS
BOGOTA
2015

CONTENIDO

Contenido

INTRODUCCION

JUSTIFICACION

OBJETIVOS

LENGUAJE DE PROGRAMACION

ANÁLISIS DEL PROGRAMA

CONCLUSIONES

BIBLIOGRAFIA

INTRODUCCIÓN

En este trabajo se manejan temas de la aplicación de las propiedades y principios de la Programación Orientada a Objetos (POO). En este proyecto que se realizó con el fin de entregarse como trabajo final de la asignatura de Programación Orientada a Objetos, se creó un juego el cual se llamó DUCK HUNT UD que se basa en la idea del videojuego creado por la multinacional Nintendo® llamado de igual manera. En dicho proyecto se trabaja con las características del paradigma POO bajo un lenguaje apto para el desarrollo del aplicativo.

JUSTIFICACIÓN

El manual sirve para facilitar el manejo y entendimiento de las características de la programación orientada a objetos, sus beneficios y aplicaciones a la hora de implementar programas y las facilidades para desarrollar un aplicativo en un lenguaje de programación orientado a objetos conocido como Java.

De igual manera se busca ampliar conocimientos acerca de la estructura y funcionamiento de la aplicación desarrollada con el interés de difundir estos conocimientos a otros usuarios que se interesen en el manejo de este programa.

LENGUAJE DE PROGRAMACION

Para este proyecto utilizamos el lenguaje de programación orientado a objetos Java, y para el desarrollo, el IDE (Integrated Development Environment) Eclipse.

Usamos dos paquetes dentro del proyecto del desarrollo del juego. El primer paquete se llamó "imágenes" y el segundo "_UI_LN".

El primero contiene ocho clases denominadas "ConduccionDatos", "Créditos", "Elección", "Juego", "Pato", "PuntajeMaximo", "Sonido" y "VentPrint".

El segundo contiene todos los archivos que se usaron para el desarrollo del juego, entre estos se encuentran archivos de extensiones .PNG, .JPEG, .GIF, .WAV

El primer paquete contiene todo lo relacionado con la GUI y la Lógica del negocio, en ella se encuentra todo el desarrollo del juego.

El segundo paquete establece la ruta de todos los archivos usados en la lógica y la interfaz del programa.

ANALISIS DE PROGRAMA

Como se evidencia en la figura 1, el proyecto Hunter, contiene dentro del src dos paquetes: “imágenes” e “_UI_LN”:

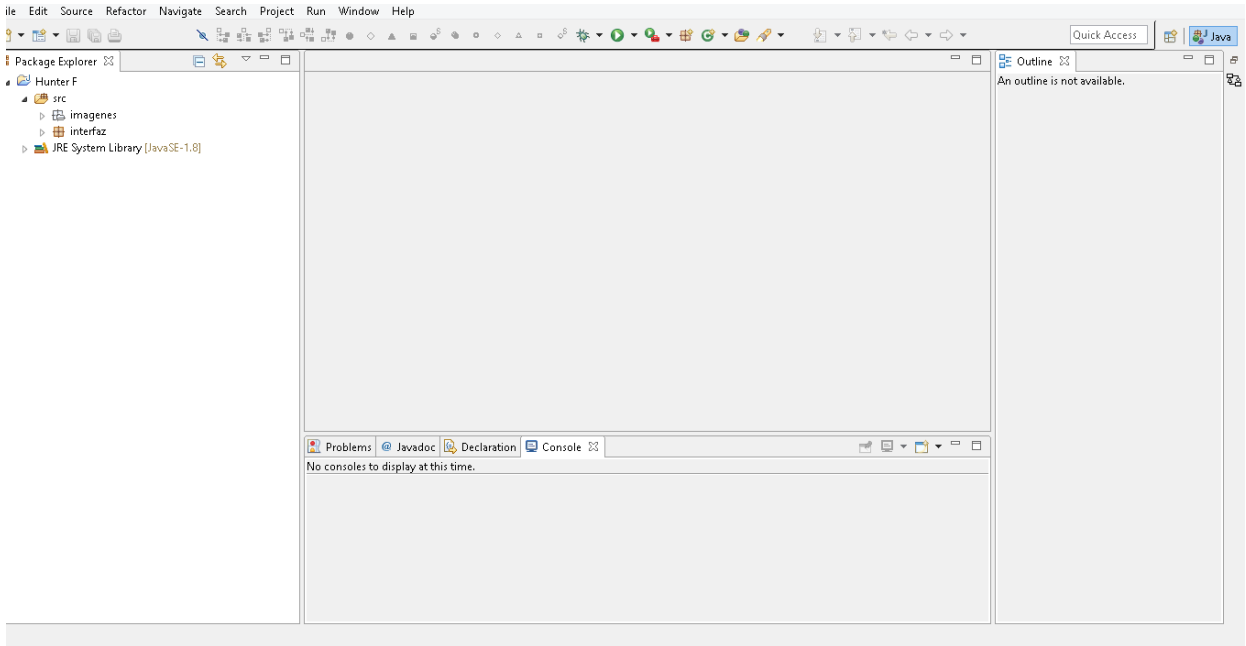


Figura 1

En la figura 2 se puede ver el contenido del paquete “imágenes”. Allí se encuentran los archivos de extensiones .PNG, .GIF, .JPEG, .WAV.

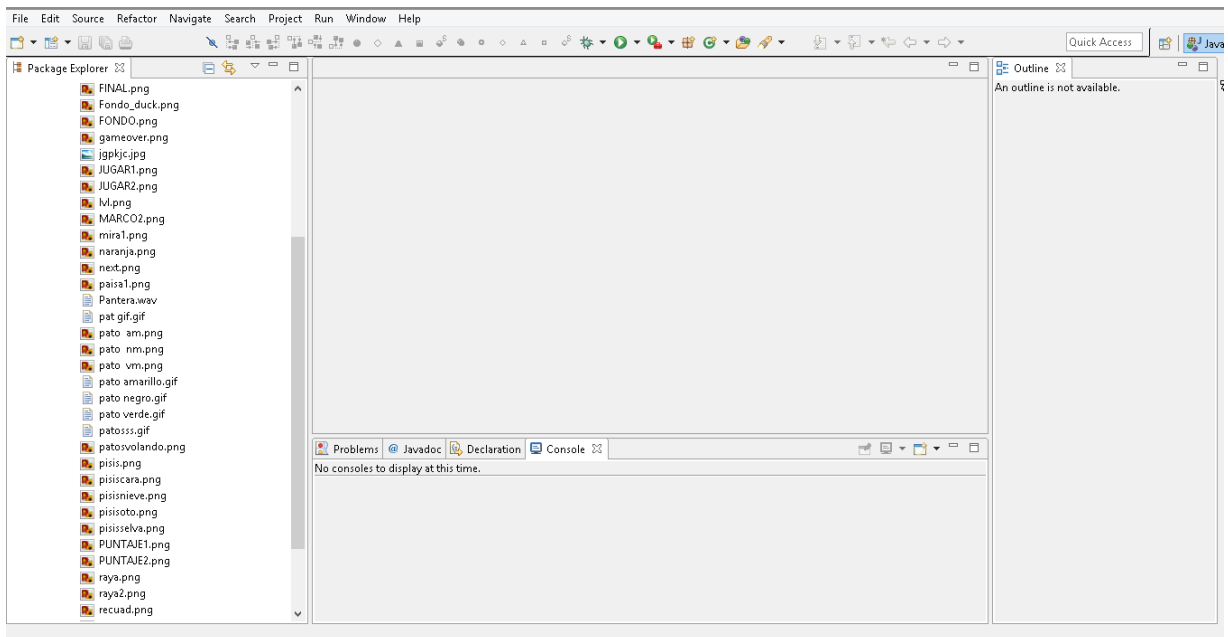


Figura 2

Ahora se evidencia en la figura 3, las clases del paquete “_UI_LN”.

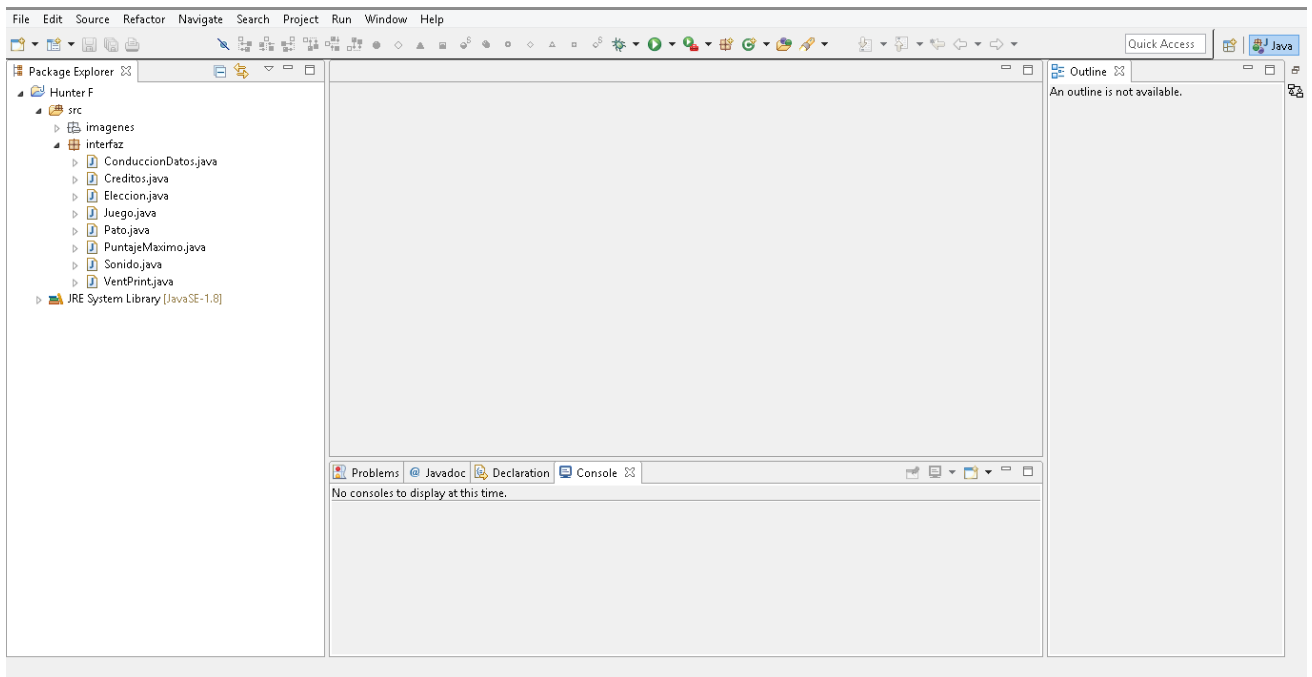


Figura 3

Nuestra ventana principal en el juego la hemos llamado VentPrint, la cual hereda de JFrame y nos permite la creación de la interfaz gráfica. En la figura 4 se puede evidenciar.

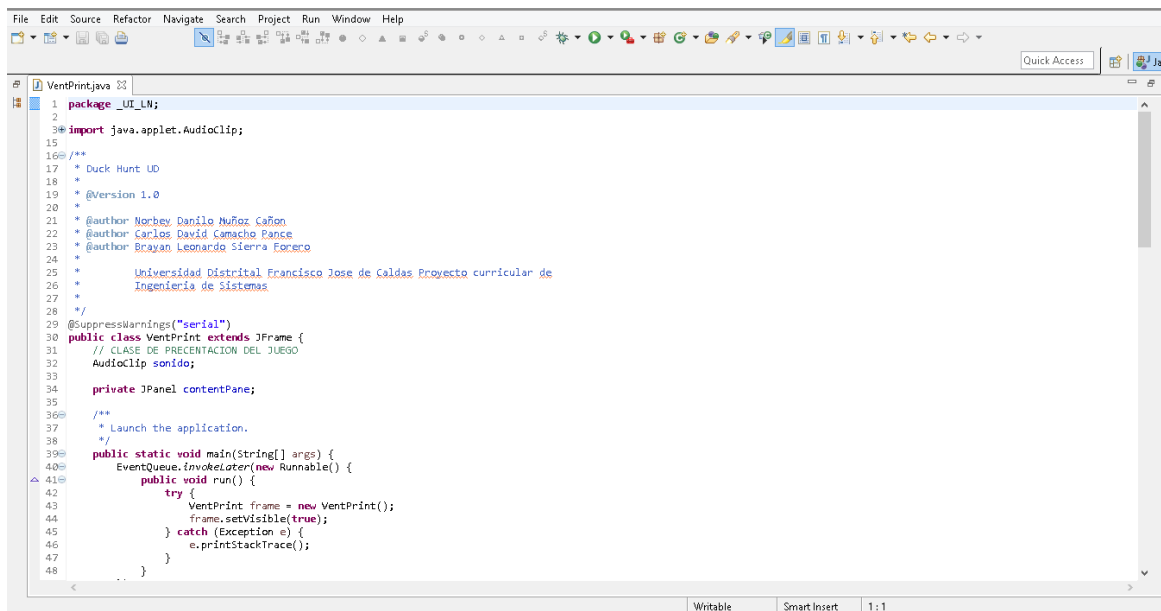


Figura 4

La GUI de VentPrint se puede ver en el la figura 5.

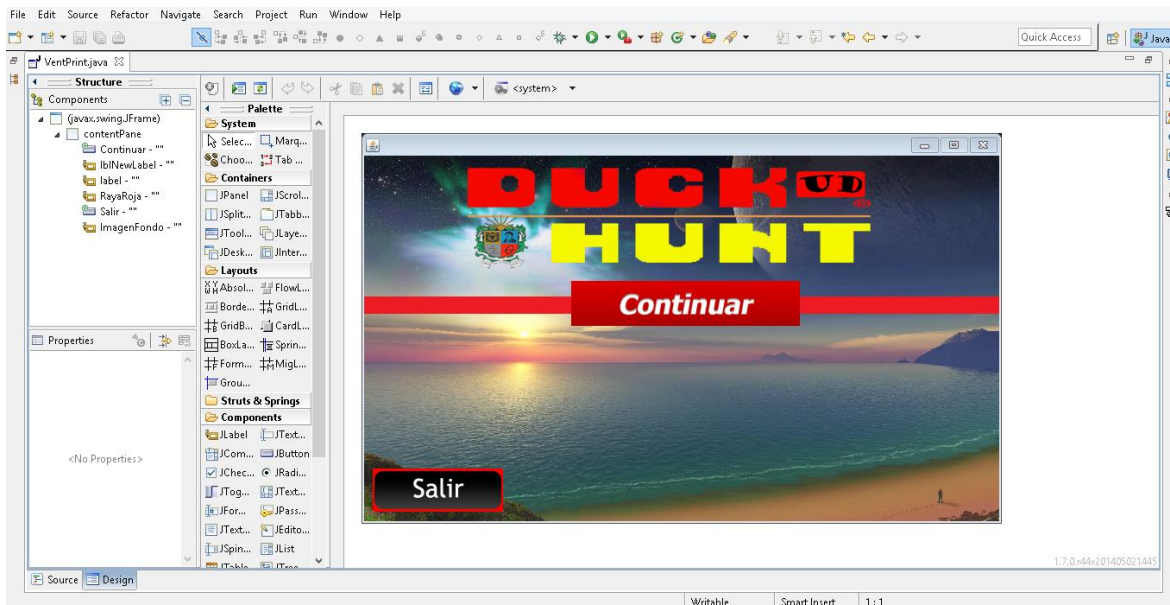


Figura 5

El botón continuar, nos direcciona a la clase que hemos llamado “Elección”, la cual nos brinda el un menú de opciones. En la figura 6 y 7 se pueden ver la Lógica y la GUI de esta clase respectivamente.

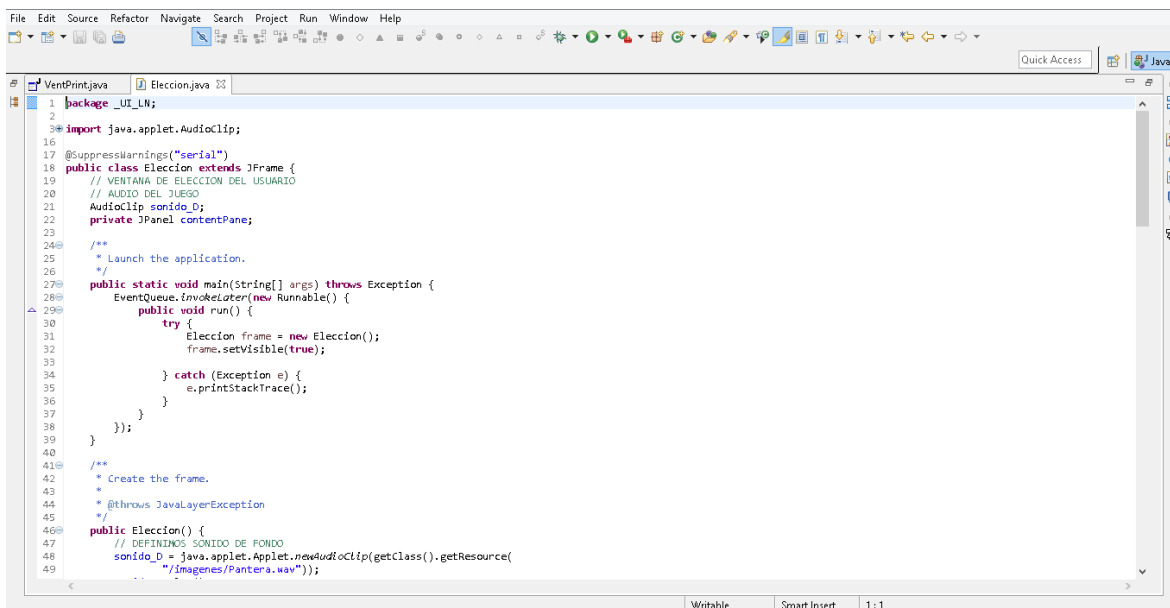


Figura 6

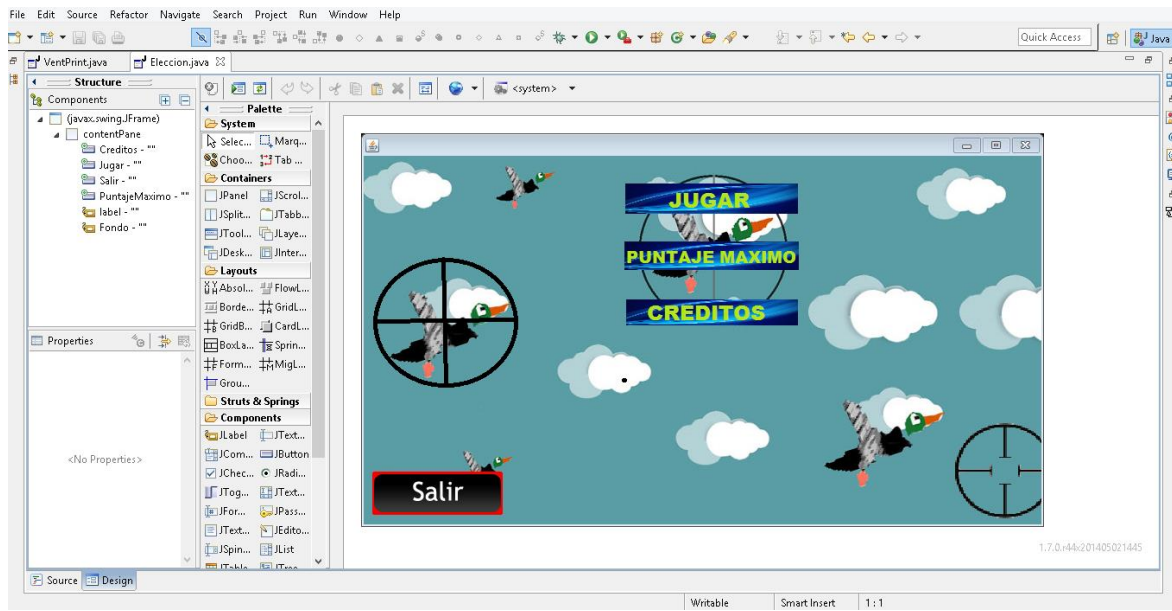
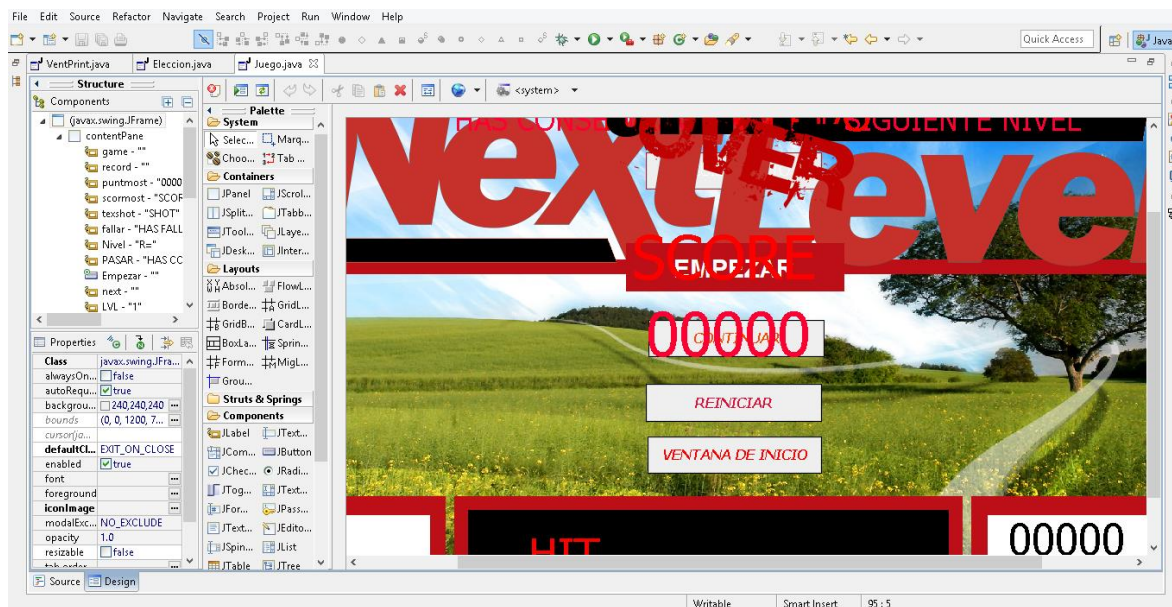


Figura 7

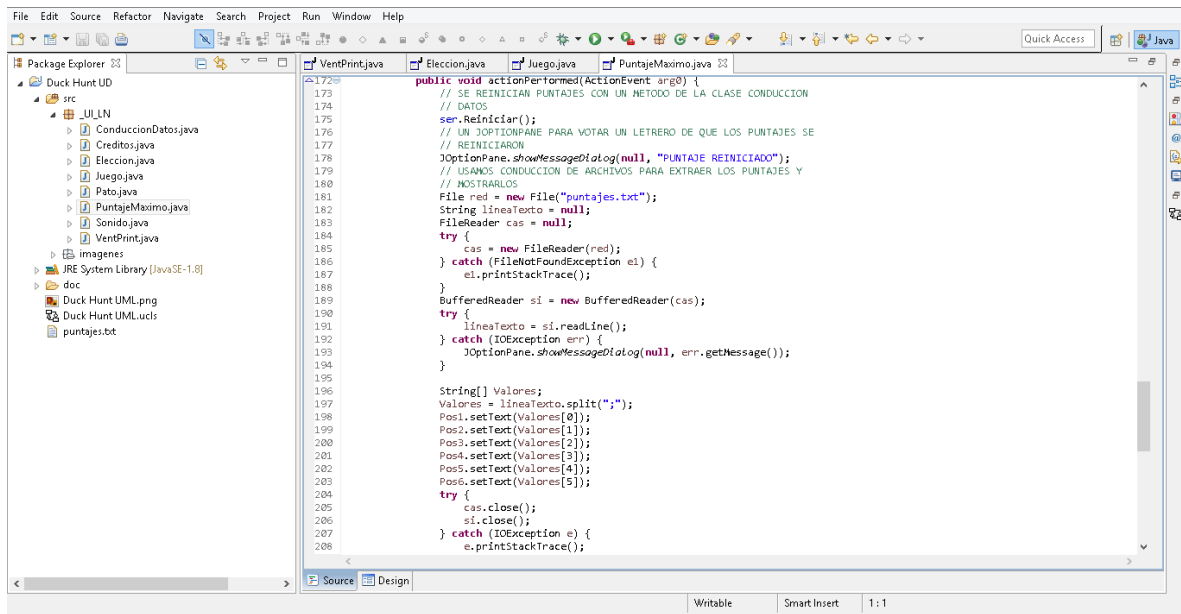
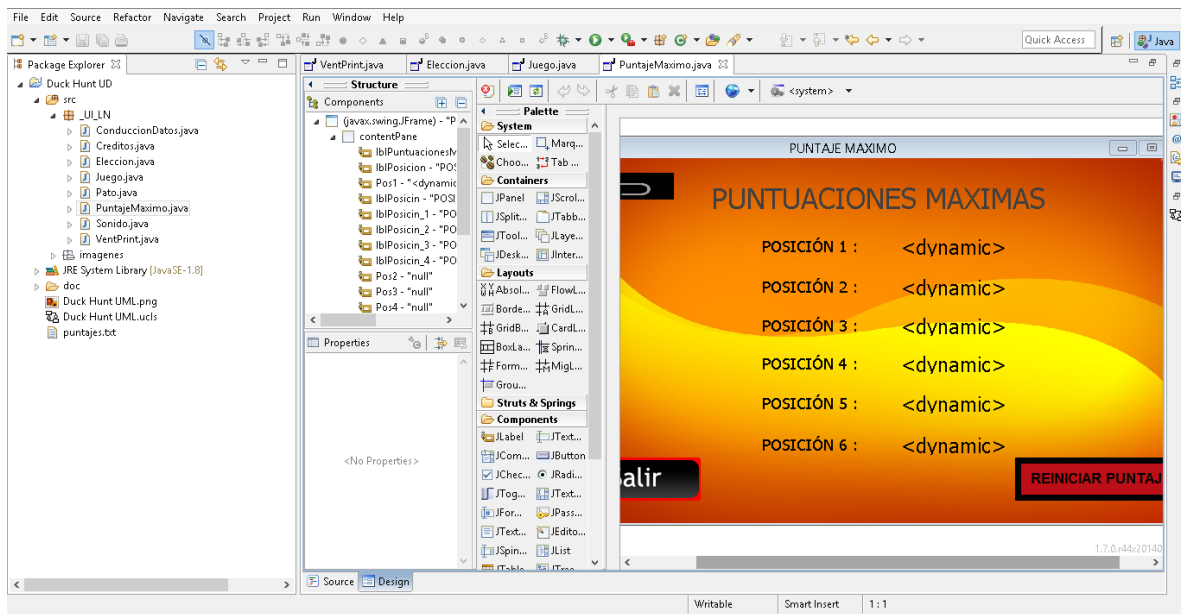
Luego se dio creación a los respectivos enlaces que brindan los botones “Créditos”, “Puntaje Máximo” y “Jugar”. Cada uno de ellos redirecciona a una clase que provee su contenido.

El botón “Jugar” nos imprime la ventana del Juego.

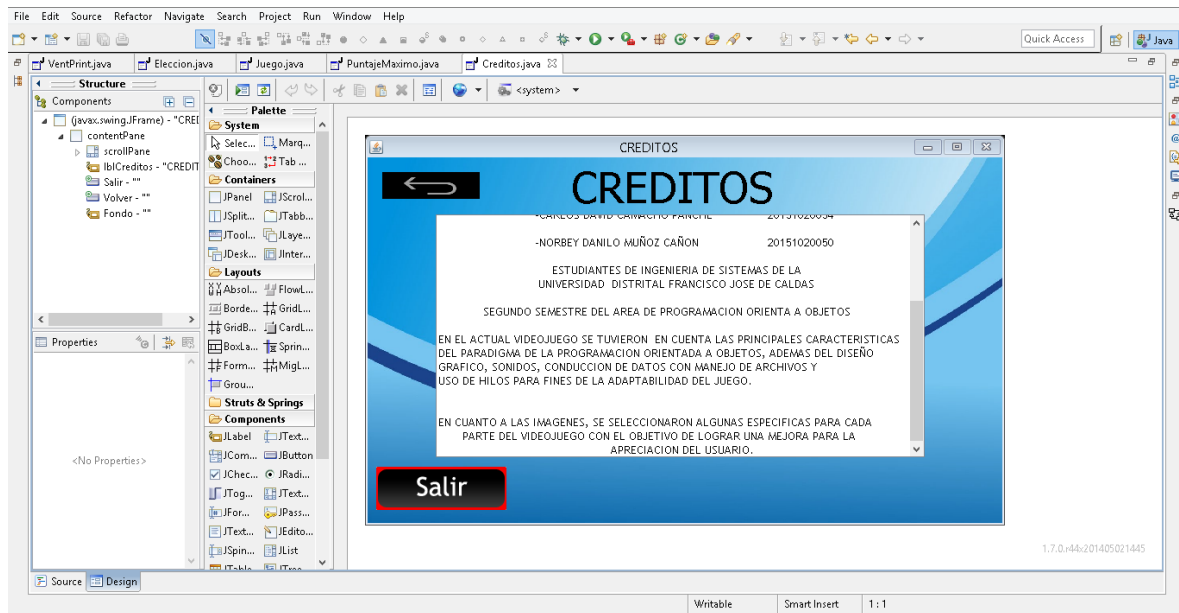


Como se puede ver, contienen todos los mensajes, botones y label's que se presentaran durante el transcurso del juego.

El botón “Puntaje Máximo”, nos muestra la ventana “PuntajeMaximo”, en este se carga un archivo .txt, que hemos llamado puntajes.txt, que contiene los records establecidos en el juego. Se imprime en una lista.



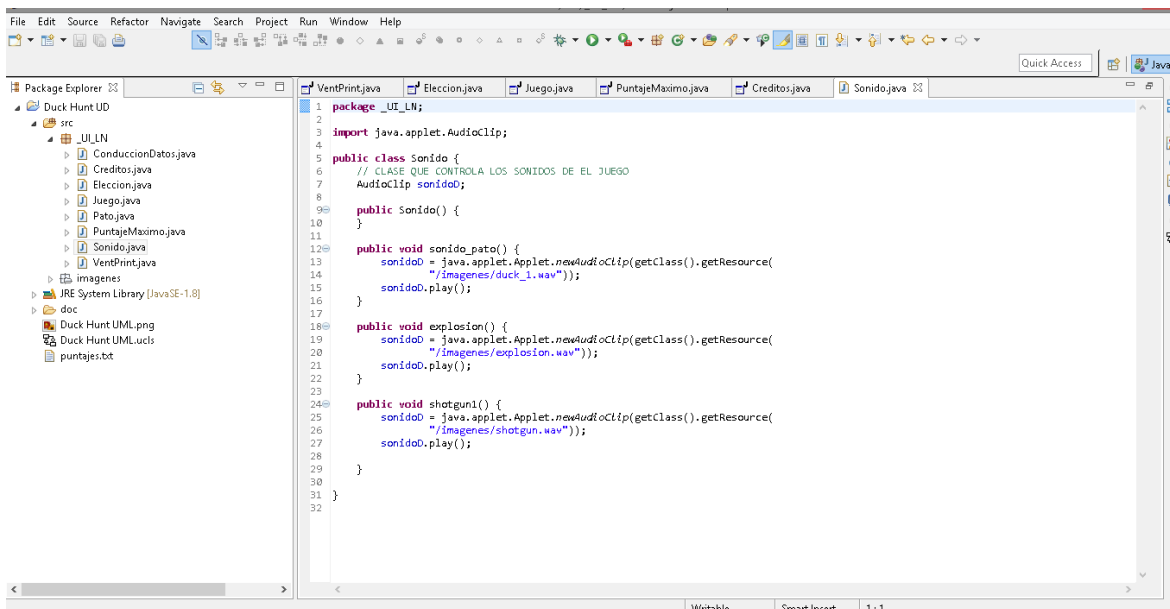
El botón Créditos, nos muestra una ventana alterna que se encarga de mostrar en un Área de texto la información sobre los desarrolladores.



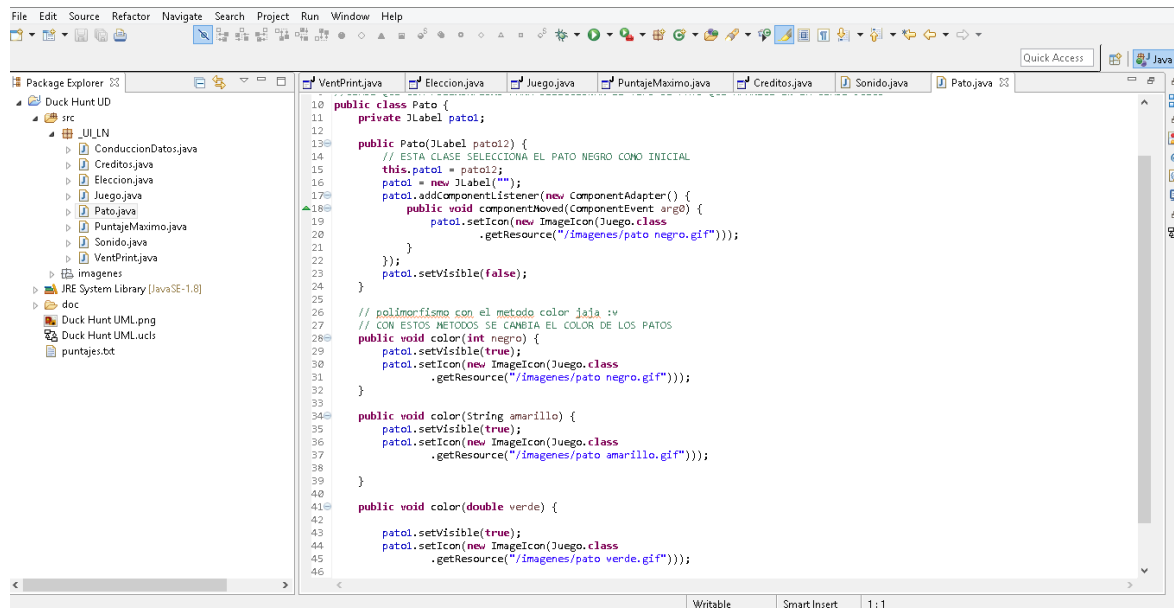
Las clases que contienen parte de la Lógica, las mostraremos a continuación, estas son “Sonido”, “Pato” y “ConduccionDatos”

La clase Sonido:

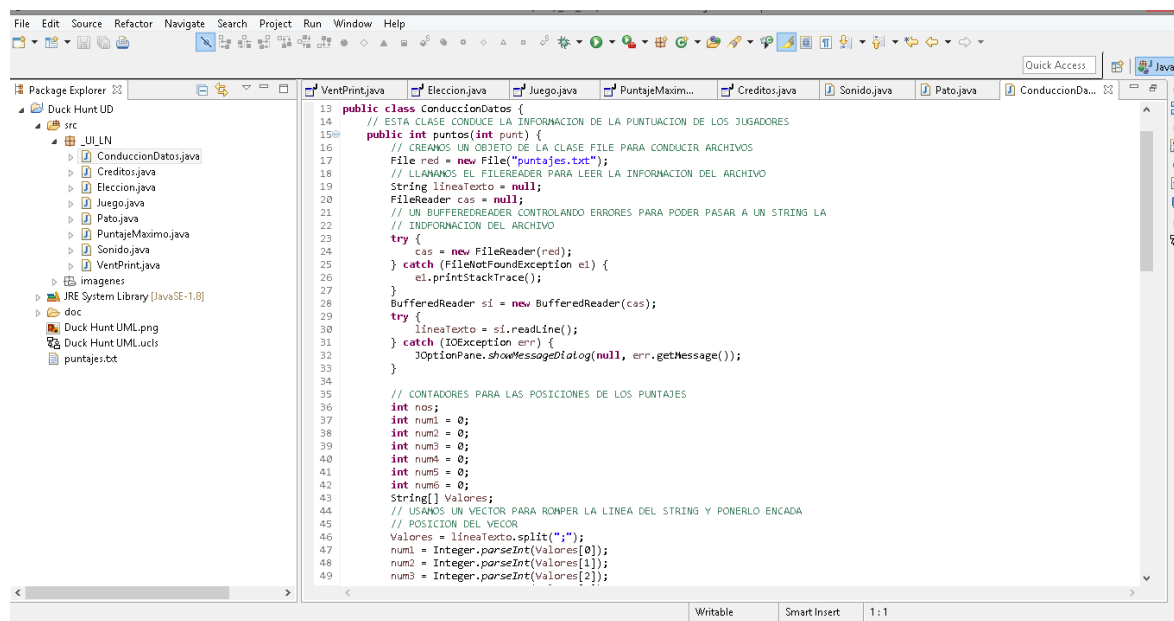
En esta clase Sonido, se cargan los audios que se utilizaran en el videojuego. Se usan las clases “AudioClip” y el “java.applet.AudioClip”. De las otras ventanas se podran acceder a los metodos, que se encargan de reproducir el audio.



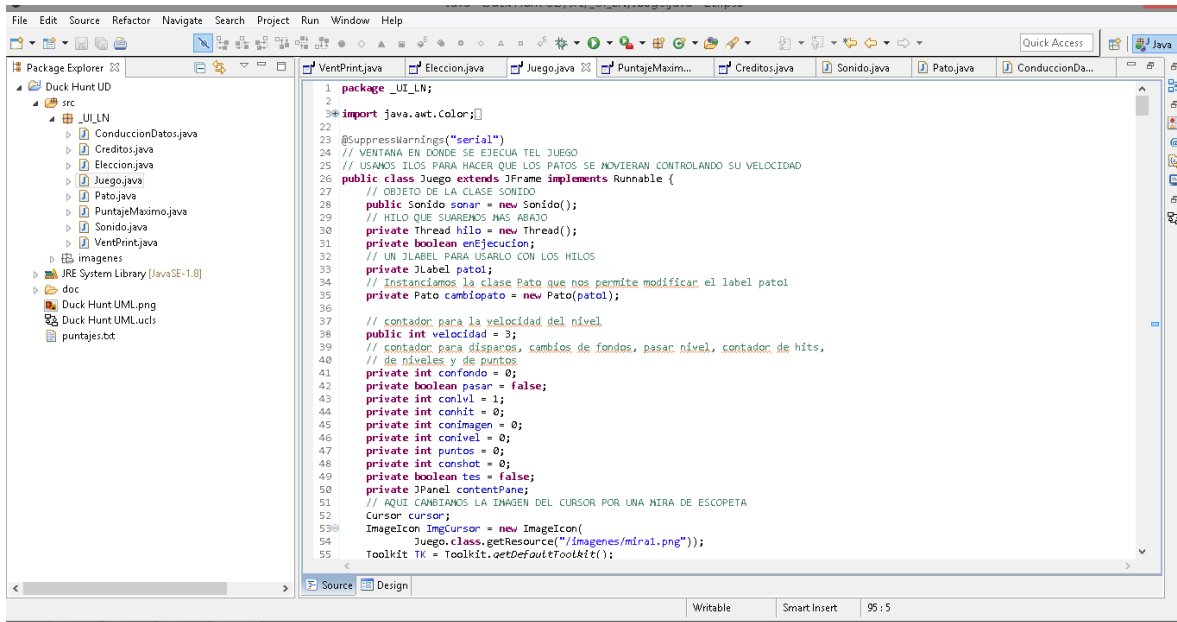
La clase “Pato” se encargara de cambiar la imagen de los patos cuando se les ejecuta la acción del Mouse: De la clase juego, se hará el llamado de sus métodos.



La clase “ConduccionDatos” se encarga de grabar los datos en un archivo de texto, estos datos serán los Máximos puntajes que se obtengan al jugar:



En la clase Juego se implementó de Runnable para el movimiento del label en el que se encuentra la imagen del pato.



CONCLUSIONES

- ❖ Obtener los resultados de los requerimientos que se plantearon inicialmente, que se visualizan satisfactoriamente en el producto final.
- ❖ El uso del IDE Eclipse y su eficacia para el desarrollo del aplicativo.
- ❖ Demostrar las capacidades y habilidades en el paradigma de la programación orientada a objetos.

BIBLIOGRAFIA

- ❖ En este trabajo no utilizamos ningún libro.
- ❖ Se usaron algunos enlaces online.